

**UNIVERSIDAD AUTONOMA DE MADRID**

**ESCUELA POLITECNICA SUPERIOR**



**Grado en Ingeniería Informática**

## **TRABAJO FIN DE GRADO**

**Creación y gestión de redes de grupos de autoayuda en países  
subdesarrollados con ayuda de la plataforma SHG**

**Julio González Calderón**

**Tutor: Denis Vaillo Sánchez**

**Ponente (si procede): Iván González Martínez**

**Enero 2018**



**Creación y gestión de redes de grupos de autoayuda en países  
subdesarrollados con ayuda de la plataforma SHG**

**AUTOR: Julio González Calderón**

**TUTOR: Denis Vaillo Sánchez**

**Dpto. de Tecnología Electrónica y de las Comunicaciones**

**Escuela Politécnica Superior**

**Universidad Autónoma de Madrid**

**Enero de 2018**





# Resumen

Este Trabajo de Fin de Grado consiste en el desarrollo de una aplicación Web, para uso real, con el objetivo de ayudar a gestionar las ayudas económicas recibidas por ciertas comunidades de países subdesarrollados a lo largo de todo el mundo.

Las diferencias entre la disponibilidad y el acceso a las tecnologías en dichos lugares con respecto a los países desarrollados es cada día mayor debido al alto índice de corrupción, criminalidad y mala gestión de dichos países. Es por ello, que uno de los objetivos principales será la creación de una red de grupos de autoayuda donde, bajo la supervisión y dirección de un facilitador, se encarguen de guiar y aconsejar a emprendedores locales con el fin de evitar que las inversiones y las ayudas económicas recibidas por dichos lugares resulten infructuosas.

Para lograrlo, dicha aplicación Web estará conectada con la plataforma Self Help Group, proporcionando distintas herramientas y métricas que, en función del rol del usuario actual de la aplicación, permitirán una variedad de acciones al usuario. Siendo algunas de las tres más importantes las siguientes:

- El manejo de la información, accediendo y añadiendo nueva información a la base de datos del sistema.
- La gestión de la información, utilizando dicha información para, por ejemplo, la creación de nuevos grupos de autoayuda.
- La visualización de la información, gracias a un mapa que permita ver la ubicación exacta y las características asociadas a cada grupo.

En el siguiente documento se recogen todas las etapas por las que ha pasado el proyecto, desde su inicio hasta su momento actual, ya que al tratarse de un proyecto real aún no está completo en su totalidad. También se incluyen una serie de anexos, donde se detallará de manera más explícita los apartados descritos anteriormente.

## Palabras clave

Aplicación Web, países subdesarrollados, grupos de autoayuda, ayudas económicas, Self-Help-Group



# **Abstract**

This Bachelor Thesis is based on the design of a web application for real use, with the aim of managing the financial aid received by certain communities in underdeveloped countries all over the world.

The differences in terms of availability and access to technology between developed and underdeveloped countries are increasing, due to the high index of corruption, criminality and poor management in the underdeveloped countries. For this reason, one of the main objectives of this project is the creation of self-help group networks, in which a facilitator supervises and gives advice to local entrepreneurs. This system will prevent inversions and financial help from being unsuccessful.

To reach this objective, the web application designed in this paper will be connected to the self-help group platform, providing different resources and metrics, which will allow the user to carry out different actions. Three of the most important actions are the following ones.

- Information management, by accessing and adding new information to the system's database
- Information management, by using that information to create new self-help groups
- Information display, by using a map which shows the exact localization and the main characteristics associated to each group.

The present paper explains the different stages of the project, from the beginning to the current moment (it is a real project which has not been finished yet). This paper also contains a series of attached documents in which the points described before, will be explained in a more detailed way.

## **Keywords**

Web application, underdeveloped countries, self-help-groups, financial aids





## ***Agradecimientos***

*A la Universidad por haberme dado la oportunidad de formarme aquí.*

*A todos mis compañeros de carrera por haberme acompañado, ayudado y apoyado durante esta etapa de mi vida.*

*A mi ponente Iván González Martínez por ayudarme con la última asignatura de la carrera y guiarme durante todo este proceso tan bien como lo ha hecho.*

*A Global Incubator por darme la oportunidad de realizar las prácticas de empresa y el TFG con ellos. En especial a Denis y a Adri por haberme ayudado tanto con el desarrollo del TFG.*

*A Eli por saber que más que una amiga, es una hermana.*

*A Carol por aparecer en mi vida sin esperarlo y formar parte de ella.*

*A mi familia por creer siempre en mí y apoyarme durante todos estos años. En especial a mi Padre y a mi Hermana por ser los dos pilares más importantes de mi vida.*

*Y sobre todo a ti Mamá, porque me viste empezar pero no me verás acabar. Estés donde estés espero que estés orgullosa. Va por ti.*



# INDICE DE CONTENIDOS

1	Introducción .....	5
1.1	Motivación.....	5
1.2	Objetivos.....	6
1.3	Organización de la memoria.....	7
2	Estado del arte.....	9
2.1	Global Incubator.....	9
2.1.1	Introducción.....	9
2.1.2	Acerca de la empresa.....	9
2.1.3	Palabras y conceptos clave.....	10
2.1.3.1	Webintra.....	10
2.1.3.2	AML.....	11
2.1.3.3	Apps .....	11
3	Análisis y Diseño .....	12
3.1	Ciclo de vida del proyecto.....	12
3.2	Diagrama de casos de uso.....	12
3.3	Análisis de requisitos.....	13
3.4	Diseño de la Base de Datos.....	16
4	Desarrollo.....	17
4.1	Herramientas de trabajo.....	17
4.1.1	Hardware.....	18
4.1.2	Software.....	18
4.1.2.1	Sistema Operativo.....	18
4.1.2.2	Webintra.....	18
4.1.2.3	QApps .....	18
4.1.2.4	FreeApps .....	19
4.1.3	Lenguajes de programación.....	19
4.1.3.1	AML.....	19
4.1.3.2	JavaScript.....	20
4.1.3.3	jQuery .....	21
4.1.3.4	HTML .....	21
4.1.3.5	CSS .....	22
4.1.3.6	AJAX .....	22

4.1.3.7 Bootstrap .....	23
4.1.3.8 SQL .....	23
4.1.4 Herramientas usadas .....	24
4.1.4.1 Google Docs & Spreadsheets .....	24
4.1.4.2 Trello .....	24
4.1.4.3 WhatsApp .....	25
4.1.4.4 Microsoft Word .....	25
4.1.4.5 Microsoft Excel .....	25
4.1.4.6 API REST .....	26
4.2 Estado inicial del proyecto .....	26
4.2.1 Diseño del Controlador .....	27
4.2.2 Diseño de la Vista .....	28
4.2.3 Diseño del Modelo .....	33
5 Integración, pruebas y resultados .....	35
6 Conclusiones y trabajo futuro .....	39
6.1 Conclusiones .....	39
6.2 Trabajo futuro .....	40
Referencias .....	41
Glosario .....	43
Anexos .....	- 1 -
A Requisitos de la aplicación .....	- 1 -
Requisitos funcionales .....	- 1 -
Comunes a Facilitador, Organización y Admin .....	- 1 -
Específicos para el Subsistema Organización .....	- 5 -
Específicos para el Subsistema Administrador .....	- 5 -
Requisitos no funcionales .....	- 8 -
Documentación .....	- 8 -
Rendimiento .....	- 9 -
Disponibilidad y accesibilidad .....	- 9 -
Seguridad .....	- 9 -
Usabilidad .....	- 9 -
Estabilidad .....	- 10 -
Portabilidad .....	- 10 -
Operatividad .....	- 10 -

	Mantenibilidad .....	- 11 -
	Interfaz .....	- 11 -
B	Feedback con el cliente.....	- 12 -
C	Pruebas realizadas.....	- 15 -
D	Sintaxis básica de AML.....	- 19 -
E	Ejemplos de tipos de AML.....	- 20 -
F	Ejemplo de uso de QApp.....	- 24 -
G	Ejemplo de uso de FreeApp.....	- 28 -
H	Parseador y ejemplos de AML.....	- 30 -

## INDICE DE FIGURAS

FIGURA 1: LA BRECHA DIGITAL .....	5
FIGURA 2: LOGO Y STARTUPS DE CAPITAL CERTAINTY .....	9
FIGURA 3: CICLO DE VIDA DE SOFTWARE ITERATIVO E INCREMENTAL.....	12
FIGURA 4: DIAGRAMA DE CASOS DE USO DE LA APLICACIÓN .....	13
FIGURA 5: DISEÑO DE LA BASE DE DATOS .....	17
FIGURA 6: MODELO VISTA CONTROLADOR .....	27
FIGURA 7: LIBRERÍAS USADAS .....	28
FIGURA 8: APLICACIÓN DE BOOTSTRAP .....	29
FIGURA 9: CÓDIGO DE APLICACIÓN DE BOOTSTRAP .....	29
FIGURA 10: USO DE JQUERY Y JS .....	30
FIGURA 11: FUNCIONES AJAX.....	31
FIGURA 12: EJEMPLO 1 DE AJAX .....	32
FIGURA 13: EJEMPLO 2 DE AJAX .....	33
FIGURA 14: API SHG .....	35
FIGURA 15: SPRINT CON LOS MIEMBROS DEL EQUIPO.....	- 12 -
FIGURA 16: CORRECCIONES DEL CLIENTE SOBRE EL DISEÑO.....	- 13 -
FIGURA 17: PATRONES DE DISEÑO DEL CLIENTE .....	- 13 -
FIGURA 18: FEEDBACK CON EL CLIENTE A TRAVÉS DE WHATSAPP .....	- 14 -
FIGURA 19: FEEDBACK CON EL CLIENTE A TRAVÉS DE TRELLO .....	- 15 -
FIGURA 20: PRUEBA LOGIN .....	- 15 -
FIGURA 21: CAMBIO DE IDIOMA .....	- 16 -
FIGURA 22: CAMBIO DE COORDENADAS .....	- 17 -
FIGURA 23: CREAR NUEVO USUARIO .....	- 18 -
FIGURA 24: MENU 1 DE QAPPS .....	- 24 -
FIGURA 25: MENÚ 2 DE QAPPS .....	- 25 -
FIGURA 26: MENÚ DE FACILITATOR.....	- 27 -
FIGURA 27: AÑADIR CUESTIONES .....	- 28 -
FIGURA 28: EJEMPLO DE USO DE FREEAPP .....	- 29 -
FIGURA 29: PARSEADOR AML .....	- 30 -
FIGURA 30: EJEMPLO 2 DE PARSEADOR AML .....	- 31 -
FIGURA 31: EJEMPLO 3 DE PARSEADOR AML.....	- 32 -

# 1 Introducción

---

## 1.1 Motivación

Hoy en día vivimos en una era en la que la tecnología crece día tras día de manera exponencial. La llamada Era Digital, que se define como el período de la historia de la humanidad que va ligado a las Tecnologías de la Información y la Comunicación (TIC). Pero dicha Era, tristemente, va de la mano con la llamada Brecha Digital, definida como la separación que existe entre las personas (comunidades, estados, países...) que utilizan las Tecnologías de Información y Comunicación (TIC) como una parte rutinaria de su vida diaria y aquellas que no tienen acceso a las mismas y que aunque las tengan no saben cómo utilizarlas.[\[1, 2, 3\]](#)

Pero nos hemos parado a preguntarnos ¿cómo viven esta Era los países subdesarrollados?, ¿cómo de grande es esa Brecha? Países donde en pleno siglo XXI todavía hay gente muriendo de hambre diariamente debido a la pobreza, en su mayoría causada por la corrupción y mala gestión económica y social de estos lugares.

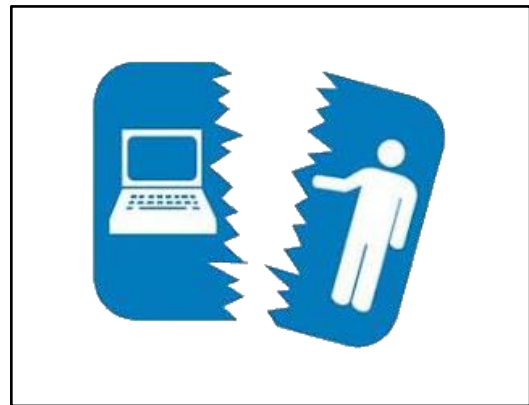


FIGURA 1: LA BRECHA DIGITAL

Parece algo frívolo pensar en nuevas tecnologías cuando aún existen personas que carecen de las necesidades básicas que todo ser humano debería de tener. Y aunque es cierto que la globalización tecnológica ha abierto nuevas oportunidades para los países subdesarrollados, pero si comparamos los progresos de la ciencia en países desarrollados con países subdesarrollados podemos ver que estos últimos son tan lentos que cada día la diferencia entre estos países se hace aún mayor, lo que provoca que tanto la dependencia de los países subdesarrollados con respecto a los desarrollados y la Brecha Digital de estos sea cada vez más y más profunda.

Para muchos de los altos cargos de los países desarrollados, el objetivo de conseguir que las TIC lleguen y se usen adecuadamente a los lugares más pobres supone todo un reto de cara a obtener algún tipo de beneficio. Y es que, hay un gran desacuerdo por parte de la opinión pública y las ONGs, que consideran un sinsentido la integración de las TIC en



países subdesarrollados, ya que lo primero es ayudar a esas personas a conseguir un trabajo, para que puedan tener un hogar y en consecuencia dichas tecnologías.

Es por todo lo mencionado anteriormente de donde surge la motivación de realizar este proyecto, para tratar de mejorar la gestión económica de dichos lugares ayudando a reducir la profunda Brecha Digital existente, ya que si se tiene un control de cómo se invierten las ayudas económicas recibidas por cada uno de estos lugares, habrá más opciones para poder invertir en nuevas tecnologías.

## ***1.2 Objetivos***

El primer objetivo por el que me planteé este proyecto como Trabajo de Fin de Grado fue a nivel personal, ya que quería ver si era capaz de aplicar los conocimientos que he adquirido durante toda la carrera. Para ello ha sido necesario tener claros conceptos de asignaturas como “Proyecto de Ingeniería del Software” e “Ingeniería del Software” para saber cómo se realiza la planificación de un proyecto real en el que se interactúa con un cliente, “Sistemas Informáticos I” para saber cómo realizar una página Web tanto a nivel de front-end como back-end y “Estructura de Datos” para saber todo lo referente a bases de datos.

Un proyecto de estas características no solo te permite aprender cómo desarrollar una aplicación Web a grandes rasgos, si no que tienes que analizar detenidamente cada paso con el cliente para saber qué es lo que realmente quiere y ver cómo esas decisiones pueden afectar a etapas futuras, a pesar de que, posiblemente, el cliente cambie de opinión en un futuro.

En cuanto al desarrollo, ha supuesto un reto importante para saber cómo mezclar y usar las tecnologías Web (AJAX, HTML, CSS, jQuery, JS, etc.). Ya que a pesar de conocer la mayoría de estas tecnologías, ha sido necesario profundizar mucho más en todas ellas.

Además, no solo he buscado realizar este proyecto para aprender cosas nuevas y afianzar conceptos pasados, si no para poder aportar mi granito de arena en esta causa intentando ayudar al mayor número de personas de estos lugares y, obviamente, poder sentirme orgulloso de ello.

### ***1.3 Organización de la memoria***

La memoria de este proyecto consta de los siguientes capítulos:

- **Capítulo 2 – Estado del arte**

En este capítulo se hablará acerca de la empresa, haciendo una breve introducción de la misma y de todas las palabras clave necesarias para poner en contexto al lector.

- **Capítulo 3 – Análisis y Diseño**

En este capítulo se mostrará el análisis previo realizado sobre el proyecto, comentando y presentando las decisiones de diseño que se han tomado.

- **Capítulo 4 – Desarrollo**

En este capítulo se hablará sobre el conjunto de herramientas, tanto software como hardware, utilizadas para el desarrollo del proyecto. También se documentará el estado inicial del proyecto y las mejoras realizadas.

- **Capítulo 5 – Integración, pruebas y resultados**

En este capítulo estarán las pruebas y las conclusiones obtenidas tras la realización de pruebas sobre los diferentes escenarios de la aplicación.

- **Capítulo 6 – Conclusiones y trabajo futuro**

Por último, en este capítulo se recogerán las conclusiones obtenidas tras la finalización del desarrollo del proyecto y se marcarán las pautas que deberán seguirse para en un futuro mejorar el proyecto.



## 2 Estado del arte

---

### 2.1 *Global Incubator*

#### 2.1.1 Introducción

Todo este TFG se ha realizado de la mano de la empresa Global Incubator, con sede en el Parque Tecnológico de la Universidad Carlos III de Leganés, Madrid.

En los siguientes apartados hablaré sobre cada uno de los distintos aspectos, características y detalles acerca de la empresa que serán necesarios comentar y conocer antes de comenzar con la explicación del proyecto, para una vez realizado el proyecto tener una visión global más adecuada sobre las mejoras y el trabajo realizado.

#### 2.1.2 Acerca de la empresa

La startup Global Incubator forma parte del conjunto de startups: Army of Change, PullApproach, Corporate Innoversity, Inner Virtuoso y Wisdom, las cuales están agrupadas dentro de la cartera de negocios de la empresa Capital Certainty.

Capital Certainty es una empresa que ha defendido el ideal de conseguir una sociedad en la que la mejora del mundo sea algo rentable. Para ello, la empresa ha invertido en distintas iniciativas que buscan tener un impacto social, además de tener una alta rentabilidad económica a través de modelos de negocio sostenibles. Su lema es que toda inversión social también puede ser rentable y toda inversión rentable, también lo puede ser social.



FIGURA 2: LOGO Y STARTUPS DE CAPITAL CERTAINTY

El proceso necesario para llevar a cabo dicho impacto comienza con la detección y focalización de todos los problemas sociales que quieren resolverse, para poder crear los

planes de negocios y la innovación de ideas necesarias que permitan su solución. Posteriormente se busca la financiación económica para llevar a cabo dichos planes y así desarrollar la tecnología necesaria.

Con la idea de apoyar a los emprendedores a llevar a cabo sus ideas para mejorar el mundo, Global Incubator proporciona todo ese entorno de trabajo que permite modelar cualquiera de los ecosistemas de innovación que los emprendedores tienen en su mente.

Esto supone, la facilitación de las herramientas necesarias para convertir una startup en una empresa, de una forma guiada y estructurada a través de un sistema de tareas, que gracias a los distintos menús, dashboards y vistas permiten saber en todo momento el estado de cada proyecto, facilitando la gestión de los mismos mediante el uso de una serie de herramientas analíticas, accesos a diferentes estudios de mercado, opiniones de expertos y por supuesto, un entorno de desarrollo con el cual desarrollaré del proyecto y comentaré en siguientes apartados.

Por lo tanto, con cada startup de Capital Certainty se aumenta el conocimiento y se pone a disposición de estrategias sociales, como en el caso de este proyecto.

Global Incubator analiza más de 240K startups, Army of Change se conecta con más de 200M de freelancers, y Corporate Innoversity virtualiza 500+ ecosistemas de innovación, para de esta forma generar conocimiento convertido en soluciones sobre la plataforma Wisdom.

### **2.1.3 Palabras y conceptos clave**

En este apartado explicaré una serie de conceptos y palabras clave relacionados con el lenguaje y las apps de trabajo del proyecto, que son exclusivos de la empresa y por lo tanto serán desconocidos para cualquier lector ajeno a la misma.

#### **2.1.3.1 Webintra**

Webintra es un entorno de desarrollo completo creado por Global Incubator que permite al usuario la subida, bajada y modificación de los archivos, el desarrollo de código y la administración de todo lo relacionado con dichos archivos y la comunicación con otros usuarios de la plataforma de una forma sencilla e intuitiva. En el Apartado 4 hablaré más detenidamente sobre esta herramienta.

### **2.1.3.2 AML**

AML es el pseudo-lenguaje de programación exclusivo de Global Incubator. Fue desarrollado mediante JavaScript y PHP por el equipo con el objetivo de satisfacer las necesidades de tener un lenguaje que permitiese a los desarrolladores de front-end acceder a funcionalidades y características propias del back-end dándole de esta forma mayor flexibilidad de desarrollo y mayor seguridad del sistema.

En el Anexo E hablaré más detenidamente sobre la sintaxis de dicho lenguaje.

### **2.1.3.3 Apps**

Webintra cuenta con una serie de módulos independientes denominados Apps que proporcionan al usuario unas funcionalidades específicas para el desarrollo de aplicaciones.

Este planteamiento permite a la empresa disponer de una gran portabilidad de sus herramientas de trabajo, hacienda que cualquier usuario pueda acceder a ellas simplemente con conexión a Internet y con los permisos de acceso oportunos.

Dentro de estas Apps mencionadas, hablaré de las 2 más importantes y las que más he utilizado para la realización del Proyecto.

#### **2.1.3.3.1 QApps**

Este tipo de App permite al usuario crear de manera rápida, sencilla, intuitiva y personal, todos los formularios que necesite para almacenar cualquier tipo de dato para la aplicación que necesite, dichos datos serán guardados en la base de datos de la empresa. En el apartado 4 hablaré más detalladamente sobre la motivación del uso de QApp y en el anexo G mostraré un ejemplo.

#### **2.1.3.3.2 FreeApps**

Este tipo de App es la que proporciona el entorno de desarrollo a Webintra y a su vez también proporciona al usuario una herramienta para la edición y desarrollo de código. En el apartado 4 hablaré más detalladamente sobre la motivación del uso de FreeApp como editor de código y en el anexo H mostraré un ejemplo.

## 3 Análisis y Diseño

### 3.1 Ciclo de vida del proyecto

Uno de los puntos más importantes antes de empezar a desarrollar una aplicación es definir el ciclo de vida del software más apropiado. Este término describe el desarrollo del software, desde la fase inicial hasta la fase final, y permite detectar errores más rápido, mejorar la calidad del software, estimar los plazos de implementación y sus costos, etc.

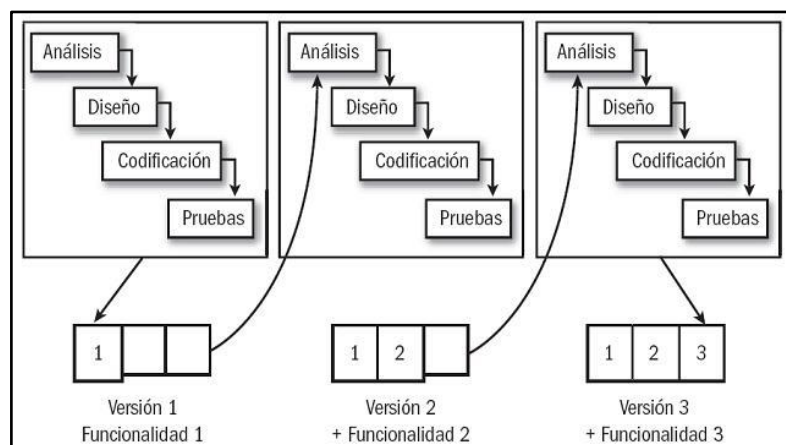


FIGURA 3: CICLO DE VIDA DE SOFTWARE ITERATIVO E INCREMENTAL

Para esta aplicación hemos optado por el ciclo de vida de software iterativo e incremental, que como su propio nombre indica consta de la fase iterativa donde en cada ciclo se revisa y se mejora el producto, y la parte incremental donde cada iteración trabaja sobre un subconjunto de la funcionalidad total pasando por las fases de Análisis, Diseño, Codificación y Pruebas, es decir, se desarrollan partes por separado para ir integrándolas a medida que se completan.

El motivo de esta elección se ha debido a que el contacto con el cliente ha sido prácticamente semanal, con cada cambio o funcionalidad realizada el cliente era avisado y él mismo probaba la nueva funcionalidad, permitiéndole incluir o desechar elementos al final de cada iteración con el fin de que el software se adapte mejor a sus necesidades reales.

### 3.2 Diagrama de casos de uso

La figura 4 muestra el diagrama de casos de uso de la aplicación. Se aprecia que hay 3 usuarios, que en orden de permisos sería: SHG Admin > Admin Organización > Facilitador,

y 2 entidades externas: API SHG y App Móvil. En el siguiente apartado se describirá en detalle la funcionalidad de cada uno de estos roles.

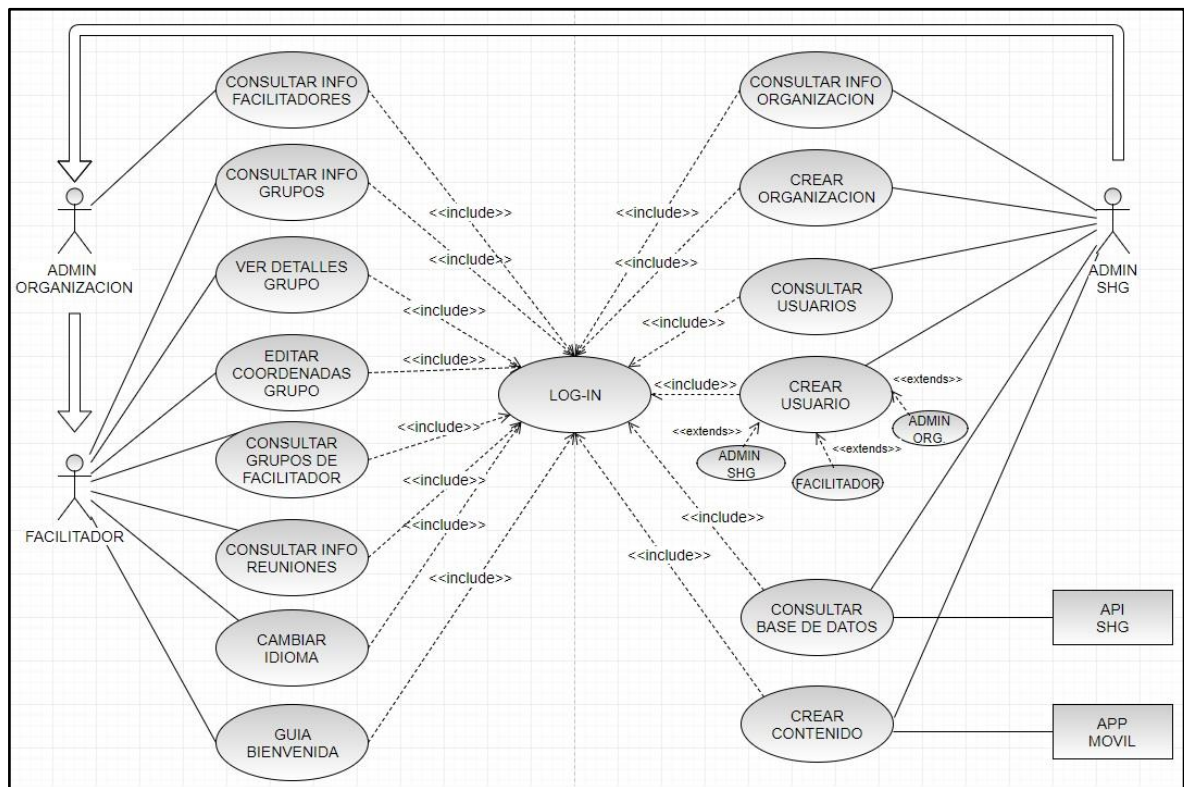


FIGURA 4: DIAGRAMA DE CASOS DE USO DE LA APLICACIÓN

### 3.3 Análisis de requisitos

Una parte fundamental en el desarrollo de un proyecto es la especificación de requisitos, que se trata de un documento que describe todas las características que debe cumplir el proyecto Web que va a ser desarrollado, con el fin de garantizar su cumplimiento antes de la finalización del mismo.

Hasta donde he implementado, el proyecto debe permitir a un usuario, de cualquier rol, el acceso a la plataforma y la interacción con la misma. Hay que decir que pese a haber acordado una especificación de requisitos previa, al tratarse de un proyecto real, se han ido manteniendo reuniones constantes con el cliente a medida que se iban desarrollando funcionalidades, para poder validar o rechazar el trabajo desarrollado.

A continuación mostraré mediante una tabla los dos tipos de requisitos, funcionales y no funcionales, que forman parte de este proyecto.



Pueden verse más detalladamente en el [Anexo A](#).

REQUISITOS FUNCIONALES		
Comunes a Facilitador, Organización y Admin		
RF 1	Iniciar sesión en la aplicación (Log-In)	El usuario debe poder iniciar sesión en la aplicación utilizando su nombre de usuario (dirección de correo electrónico) y su contraseña.
RF 2	Cierre de sesión (Log-Out)	El usuario podrá cerrar sesión en la aplicación tras pulsar el botón y confirmar su salida.
RF 3	Cambio de idioma	El usuario podrá cambiar el idioma de la aplicación en cualquier momento.
RF 4	Guía de bienvenida	El usuario podrá ver una guía rápida de uso de la aplicación.
RF 5	Consultar información de grupos	El usuario actual podrá consultar la información asociada a cada grupo.
RF 6	Ver un grupo en detalle	El usuario actual, tras acceder a la información de grupos, podrá ver información más detallada de cada grupo.
RF 7	Editar las coordenadas de un grupo	El usuario actual, tras acceder a la información de grupos, podrá editar las coordenadas de cada grupo.
RF 8	Consultar grupos de un facilitador	El usuario actual, podrá ver qué grupos tiene asociado un facilitador.
RF 9	Consultar información de reuniones	El usuario podrá consultar la información asociada a cada una de las reuniones realizadas.

Específicos para el Subsistema Organización		
RF 10	Consultar información de los facilitadores	El usuario podrá consultar la información asociada a cada una de los facilitadores.
Específicos para el Subsistema Admin		
RF 11	Consultar información de los facilitadores	El Admin podrá consultar la información asociada a cada una de los facilitadores.
RF 12	Consultar información de las organizaciones	El Admin podrá consultar la información asociada a cada una de las organizaciones.
RF 13	Crear nuevas organizaciones	El Admin podrá crear nuevas organizaciones.
RF 14	Consultar información de los usuarios	El Admin podrá consultar la información asociada a cada una de los usuarios.
RF 15	Crear nuevos usuarios	El Admin podrá crear nuevas usuarios.
RF 16	Consultar Base de Datos	El Admin podrá consultar la base de datos de la aplicación.

REQUISITOS NO FUNCIONALES	
Documentación	
RNF 1	Documentación del Sistema completo
Ayuda al usuario	
RNF 2	Ayuda al usuario
Rendimiento	
RNF 3	Rapidez de navegación
RNF 4	Soporte para todos los usuarios
Disponibilidad y accesibilidad	
RNF 5	Disponibilidad 24/7
Seguridad	
RNF 6	Claves de acceso privadas

RNF 7	Seguridad en el sistema
<b>Usabilidad</b>	
RNF 8	Recordar cuenta de usuario en terminales
RNF 9	Heurísticas de Nielsen
RNF 10	Idiomas
<b>Estabilidad</b>	
RNF 11	Aplicación consistente
<b>Portabilidad</b>	
RNF 12	Aplicación multiplataforma
<b>Operatividad</b>	
RNF 13	Operatividad adaptada
<b>Mantenibilidad</b>	
RNF 14	Mantenimiento y modificaciones
<b>Interfaz</b>	
RNF 15	Sencilla e intuitiva

### ***3.4 Diseño de la Base de Datos***

En este apartado se mostrará la estructura de la base de datos usada en la aplicación que, como se ha explicado anteriormente, la conexión de la base de datos con la lógica de la aplicación se lleva a cabo mediante el uso de AML (basado en PHP) usando la librería de mySQL y los métodos de la API REST.

Como se puede apreciar en la Figura 5 la mayoría de las tablas están conectadas entre sí, por lo que, para modelar este problema, se ha usado un modelo relacional de datos. Este tipo de base de datos recibe dicho nombre porque los datos son almacenados en forma de “relaciones” o de listas de datos, que para entendernos mejor, es lo que llamamos habitualmente “Tablas”.

Para poder establecer una relación entre dos tablas es necesario incluir en una de ellas la Clave Primaria (PK) de la otra, lo que se conoce como Clave Externa (FK). Por ejemplo, en el diagrama de la figura 5 podemos ver que la Clave Primaria de la tabla

“facilitator” es “facilitator\_no” (indicada en color verde) y ésta a su vez es la Clave Externa de la tabla “group”, pudiendo de este modo relacionarse ambas tablas.

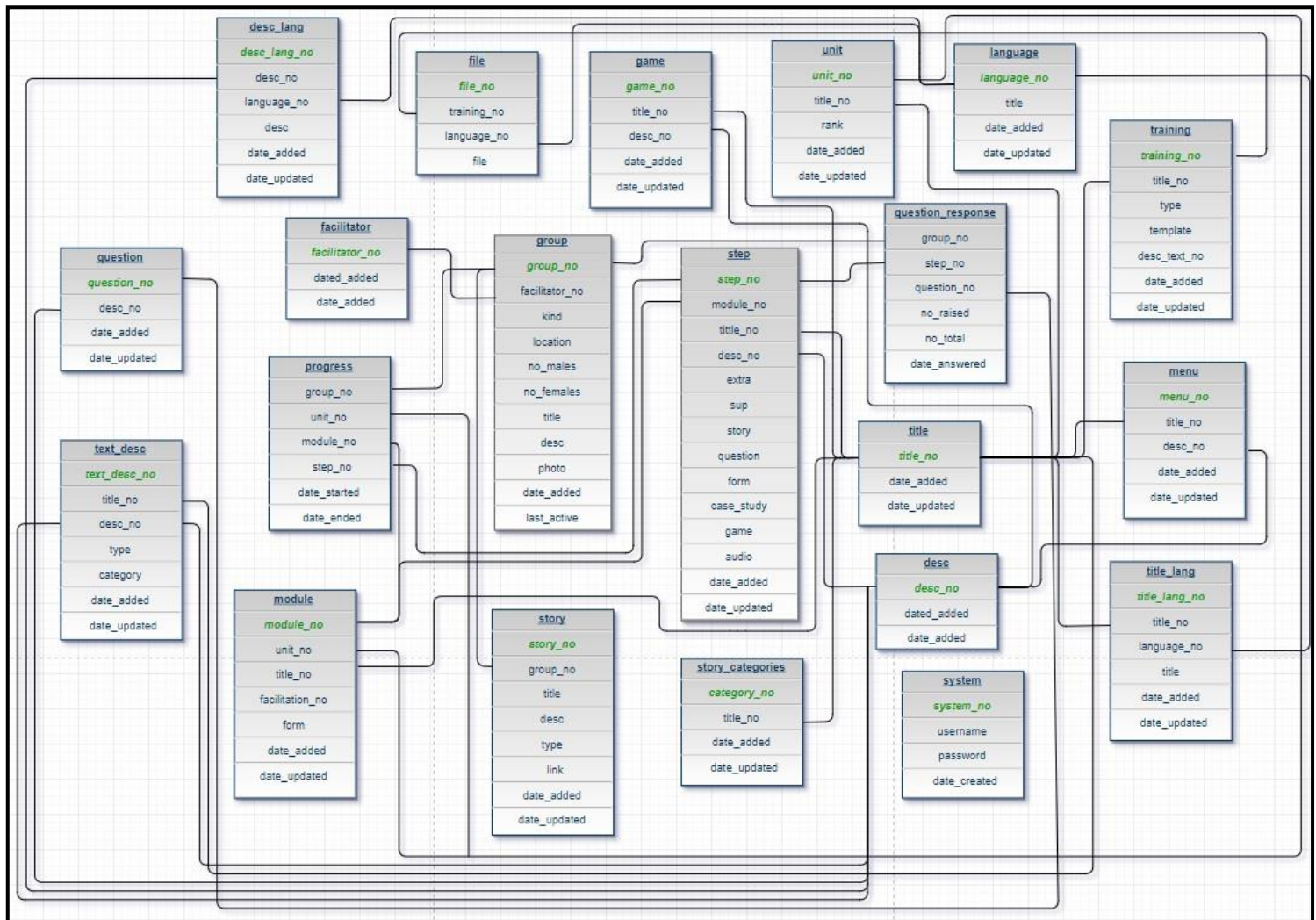


FIGURA 5: DISEÑO DE LA BASE DE DATOS

## 4 Desarrollo

### 4.1 Herramientas de trabajo

Para la realización del proyecto he usado varias herramientas y lenguajes de programación que ya conocía previamente por la Universidad pero también ha habido otros en los cuales he necesitado formarme por mi cuenta para poder realizar las funcionalidades necesarias.

### **4.1.1 Hardware**

El proyecto lo he realizado tanto en el portátil que tenía asignado en la empresa como en mi ordenador personal cuando trabajé en casa, ya que Webintra permite el acceso desde cualquier dispositivo simplemente con tu usuario y contraseña y conexión a Internet.

También he usado mi teléfono móvil personal ya que se han utilizado aplicaciones de mensajería para realizar el feedback con el cliente.

### **4.1.2 Software**

#### ***4.1.2.1 Sistema Operativo***

El trabajo realizado con el portátil de empresa fue usando Debian, el cual desconocía ya que nunca antes había trabajado con él, y he de decir que ha sido fácil y rápido adaptarse a él.

Por otro lado, en mi casa he usado Windows 10, que para el volcado y tratado de la base de datos y la realización de la memoria me ha sido mucho más útil ya que me resulta mucho más cómodo trabajar con Microsoft Word y Microsoft Excel.

#### ***4.1.2.2 Webintra***

Este entorno es mucho más completo que el que ofrecen otras plataformas de desarrollo actuales, ya que muchas de ellas apenas incluyen más de un área de texto, o en los que directamente hay que subir el código ya desarrollado sin ofrecer la posibilidad de ningún tipo de edición o desarrollo.

Es por esto que he usado este entorno de desarrollo para la realización del proyecto, ya que ofrece muchas más posibilidades que la mayoría de los competidores.

Además Webintra cuenta con funcionalidades muy útiles e interesantes como su propio chat interno para comunicar a todos sus usuarios, su muro de notificaciones para compartir noticias y su dinámica de grupos de trabajo para agrupar a miembros de un mismo proyecto.

#### ***4.1.2.3 QApps***

Se ha optado por el uso de esta App ya que es la herramienta con la que Global Incubator realiza todos sus proyectos. Esto es debido a que son unas herramientas muy útiles y versátiles y que cuando se comienza a desarrollar cualquier tipo de proyecto

pueden usarse de manera conjunta, pues un QApp puede tener como campo una pregunta de tipo QApp y permitir de este modo la relación de QApps entre sí, y en consecuencia, administrar una mayor cantidad de información de forma eficiente.

Cabe decir que los QApp son usados en grandes e importantes proyectos con clientes como Everis y Banco Santander, por lo que su rendimiento y fiabilidad es bastante alto.

#### **4.1.2.4 FreeApps**

A través de esta herramienta se realizará la mayor parte del desarrollo y codificación del proyecto. Se ha optado por usarla ya que soporta diversos lenguajes web como HTML, CSS, AML, JavaScript, JQuery y nos permite trabajar con todos ellos simultáneamente en un mismo fichero facilitándonos el desarrollo de código y vistas web.

Cada fichero de código que se cree será denominado FreeApp sin importar el tipo de código o códigos que contenga, y se le asignará un ID único para su identificación.

### **4.1.3 Lenguajes de programación**

Para añadir, mejorar e implementar todas las funcionalidades que he realizado sobre el proyecto he necesitado trabajar con una serie de lenguajes de programación, la mayor parte de ellos han sido lenguajes que normalmente están orientados al desarrollo Web.

En este apartado analizaré dichos lenguajes individualmente y comentaré como fue mi toma de contacto con cada uno de ellos, ya que algunos ya me eran familiares, otros completamente nuevos pero conocidos y otros tanto nuevos como desconocidos.

#### **4.1.3.1 AML**

Como comenté anteriormente, AML es el lenguaje propio de la empresa que aporta a Global Incubator una exclusividad total a la hora del desarrollo de sus aplicaciones. AML permite realizar unas llamadas predefinidas y muy controladas de forma que sólo se permite el acceso a aquellas partes del sistema que los desarrolladores de la plataforma deseen.

Este pseudo-lenguaje también permite el acceso a elementos propios del sistema como cuestionarios (QApps) o archivos de código (FreeApps) a través de su ID. De este modo y mediante una sencilla línea de código, se evita acceder a la base de datos del sistema, lo que podría suponer riesgos para el sistema. Es por ello que AML ha sido una pieza fundamental en el desarrollo del proyecto.

AML se forma mediante sentencias individuales que pueden ser anidadas entre sí y acopladas en cualquier tipo de lenguaje de programación conocido y soportado por WebIntra.

#### ***4.1.3.1.1 Sintaxis de AML***

Para facilitar al lector la comprensión del lenguaje AML, podemos usar la similitud de la sentencia AML con las sentencias de nuestro lenguaje natural, donde el *AML* sería el *verbo*, el *sistema* sería el *sujeto*, la palabra clave sería el *complemento directo* y los *atributos* los *complementos circunstanciales*.

Por lo tanto, en una sentencia AML le estaríamos diciendo al *sistema* que realice una determinada acción sobre un *elemento* concreto con unos determinados *atributos*.

En el Anexo D se muestran una serie de ejemplos sobre el la sintaxis de AML.

#### ***4.1.3.1.2 Tipos de AML***

Actualmente la empresa cuenta con un total de 122 tipos/sentencias de AML, número que está en constante crecimiento. Cada uno de estas sentencias de código AML tiene una funcionalidad específica y por lo tanto puede tener unos valores y atributos predeterminados, pudiendo generar distintas salidas con su ejecución (números, arrays, código HTML, JSON, AML, etc).

Por todo esto, el aprendizaje de AML ha supuesto una de las partes de mayor desafío y dificultad a la hora de realizar el proyecto, ya que (obviamente), mis conocimientos sobre este lenguaje eran completamente nulos y la documentación de la cual disponía la empresa era prácticamente inexistente y de mala calidad.

#### ***4.1.3.2 JavaScript***

Junto a AML, JavaScript ha sido el lenguaje más utilizado en el desarrollo del proyecto. Javascript es un lenguaje de programación interpretado y orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico, que se utiliza principalmente en su forma del lado del cliente (*client-side*), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. [\[4\]](#)

Dicho lenguaje ya lo conocía previamente gracias a la asignatura Sistemas Informáticos 1 cursada durante la carrera, pero debido al paso del tiempo desde la

realización de la misma tuve que refrescarme la memoria mediante tutoriales web, ya que gran parte del código que la empresa tenía previamente desarrollado estaba escrito en dicho lenguaje.

#### 4.1.3.3 *jQuery*

Dentro de JavaScript, JQuery ha sido la librería con la que más he trabajado, ya que me ha permitido la interacción con los distintos elementos del DOM de manera fácil e intuitiva. El DOM permite el acceso dinámico a través de la programación para acceder, añadir y cambiar dinámicamente contenido estructurado en documentos con lenguajes como JavaScript. [\[6\]](#)

Pese a ir de la mano con JavaScript y haber oído alguna vez hablar de él, este lenguaje no lo había visto previamente ni durante la carrera ni por cuenta propia, por lo que una de las tareas que tuve que realizar, aparte del aprendizaje de AML, fue realizar una serie de cursos/tutoriales online para poder entender los principios básicos de jQuery y así poder entender las funcionalidades que la empresa ya tenía desarrolladas. Lo que más me ha costado ha sido aprender la lógica de las funciones para buscar elementos e interactuar por el DOM.

JQuery es una biblioteca multiplataforma de JavaScript que permite simplificar la manera de interactuar con elementos de HTML con el código JavaScript. La forma de interactuar con la página es mediante la función `$()`, un alias de `jQuery()`, que recibe como parámetro una expresión CSS o el nombre de una etiqueta HTML y devuelve todos los nodos (elementos) que concuerden con la expresión. Esta expresión es denominada **selector**. [\[7\]](#)

Las principales características de esta biblioteca son

- Selección de elementos DOM.
- Interactividad y modificaciones del árbol DOM.
- Manipulación de la hoja de estilos CSS

#### 4.1.3.4 *HTML*

HTML viene del inglés *HyperText Markup Language* (lenguaje de marcas de hipertexto), y hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código



(denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. [8]

Para la realización de este proyecto ha sido fundamental saber programar en dicho lenguaje, ya sea para poder satisfacer los requisitos de diseño que el cliente nos había solicitado como para entender el lenguaje HTML correctamente y poder realizar las funcionalidades oportunas a través del uso de jQuery y JS.

Respecto a los conocimientos de este lenguaje también conocía las bases gracias a la asignatura Sistemas Informáticos 1 que cursé durante la carrera, pero he de decir que al igual que con JavaScript, estaban bastante olvidados ya y también he requerido de un pequeño aprendizaje durante el desarrollo del proyecto para ponerme al día sobre las nuevas etiquetas del lenguaje.

Además HTML es un lenguaje en constante crecimiento y renovación y actualmente se usa HTML5, el cual no existía en el momento en que cursé la asignatura mencionada.

#### **4.1.3.5 CSS**

CSS viene del inglés **Cascading Stylesheet** (Hojas de estilo en cascada) es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de las páginas web e interfaces de usuario escritas en HTML. [9]

Al igual que HTML, ha sido fundamental para el desarrollo del proyecto ya que se podría decir que sin HTML no hay CSS y viceversa, ya que uno se “complementa” al otro. Dicho lenguaje también se vio en la asignatura Sistemas Informáticos 1, pero también fue necesario realizar un pequeño aprendizaje para ponerme al día sobre la versión actual CSS3 y poder satisfacer todos los requisitos del cliente y adaptarlo a sus necesidades, ya que por muy buena que sea la aplicación que se desarrolle, si la experiencia de usuario no es buena el usuario dejará de usarla y buscará otras opciones.

Además está a la orden del día realizar las aplicaciones responsivas, lo que se conoce como Responsive Web Design es decir, adaptar adecuadamente el contenido de cada página Web en cualquier dispositivo donde se utilice, y es ahí donde entra en juego el framework Bootstrap que explicaré a continuación.

#### **4.1.3.6 AJAX**

Es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras

se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones. [\[10\]](#)

#### **4.1.3.7 Bootstrap**

Bootstrap es un framework o conjunto de herramientas de Código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales.[\[11\]](#)

El uso de este framework también ha tomado un papel importante en el desarrollo del proyecto, sobre todo en lo que diseño se refiere, ya que gracias a él ha resultado mucho más fácil la creación de las distintas secciones de cada vista Web. Combinado con CSS, Bootstrap hace muy sencillo la creación y el diseño del contenido Web.

Una de las características más destacadas de Bootstrap, y la que lo hace tan especial, es su sistema de rejillas. Dicho sistema se basa en la creación o disposición del contenido de nuestra web dentro de rejillas flexibles, las cuales se escalarán al tamaño y posición adecuada de forma automática dependiendo del tamaño de la pantalla en la que se renderice

El sistema de rejilla está pensado para ayudarnos en la disposición de los contenidos de nuestra web y su adaptación a los diferentes tamaños de pantalla de forma automática. Para ello tenemos que poner el contenido dentro de celdas o columnas que irán dentro de filas. Cada fila se puede dividir hasta en 12 columnas, pero seremos nosotros los que definiremos el número de columnas deseado para cada tamaño de pantalla [\[12\]](#). El uso de dicho sistema de rejillas ha sido muy importante a la hora de adaptar el diseño del cliente para cada tipo de dispositivo.

Respecto al conocimiento sobre este Framework también me era desconocido, ya que ni lo había visto previamente en la carrera ni había oído hablar de él hasta la llegada a la empresa. Por lo tanto también fue necesario un aprendizaje previo de dicha herramienta.

#### **4.1.3.8 SQL**

SQL (por sus siglas en inglés Structured Query Language; en español lenguaje de consulta estructurada) es un lenguaje específico del dominio que da acceso a un sistema

de gestión de bases de datos relacionales que permite especificar diversos tipos de operaciones en ellos.

Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar, de forma sencilla, información de bases de datos, así como hacer cambios en ellas. [14]

El uso de este lenguaje me ha servido para poder interpretar la base de datos del cliente y de ese modo poder volcar dicha información a los diferentes QApps.

Dicho lenguaje se vio en la asignatura EDAT y Sistemas Informáticos 1, pero también fue necesario realizar un pequeño aprendizaje para ponerme al día y poder comprender correctamente el contenido y la estructuración de las tablas de la base de datos. En el [apartado 3.4](#) se habló más en detalle sobre la base de datos de la aplicación.

#### **4.1.4 Herramientas usadas**

##### **4.1.4.1 Google Docs & *Spreadsheets***

Google Documentos y Hojas de cálculo, oficialmente Google Docs & Spreadsheets es un programa gratuito basado en Web para crear documentos en línea con la posibilidad de colaborar en grupo. Incluye procesador de textos, Hoja de cálculo, programa de presentación básico, creador de dibujos y editor de formularios destinados a encuestas [15]

Esta herramienta me ha sido de gran ayuda para mantenerme en contacto tanto con el cliente, realizando feedbacks con él sobre el diseño de la aplicación como para realizar sprints con los miembros del equipo que han participado en distintas fases del proyecto. En el [Anexo B](#) puede verse el feedback con el cliente a través de esta herramienta.

##### **4.1.4.2 Trello**

Trello es un software de administración de proyectos con interfaz web, cliente para iOS y android para organizar proyectos, que ha servido de gran ayuda para llevar un feedback constante con el cliente. En el [Anexo B](#) puede verse el feedback con el cliente a través de esta herramienta. [16]

#### **4.1.4.3 WhatsApp**

WhatsApp es una aplicación de mensajería instantánea para teléfonos inteligentes, que envía y recibe mensajes mediante Internet, complementando servicios de correo electrónico, mensajería instantánea, servicio de mensajes cortos o sistema de mensajería multimedia. [\[17\]](#)

Esta herramienta ha sido una de las más utilizadas, pues se creó un grupo del proyecto para que el cliente pudiese tener un feedback rápido, sencillo y prácticamente semanal sobre el estado del proyecto. En el Anexo B puede verse el feedback con el cliente a través de esta herramienta.

#### **4.1.4.4 Microsoft Word**

Creo que hoy en día serán pocas las personas que desconozcan una de las herramientas del paquete Microsoft Office más importantes a nivel mundial para el procesamiento de textos.

En mi caso, y pese a haber realizado la mayor parte del desarrollo del proyecto con el sistema operativo Debian, opté por dicha herramienta (y en consecuencia, Windows 10 como sistema operativo) para la elaboración de toda la documentación del proyecto, ya que es sin duda la herramienta que conozco desde hace más tiempo y con la que más cómodo me encuentro trabajando a la hora de documentar. [\[18\]](#)

#### **4.1.4.5 Microsoft Excel**

Esta herramienta ha jugado un papel muy importante a la hora de tratar toda la información que nos proporcionó el cliente para el diseño de la base de datos del proyecto. [\[19\]](#)

Gracias a Microsoft Excell fue más fácil la modificación de las columnas necesarias para organizar la información de manera correcta para posteriormente poder volcarlos en su QApp correspondiente.

Obviamente, conocía ya esta herramienta pero desconocía el gran alcance que ofrece para realizar consultas y realizar acciones sobre la información contenida, ya que no había visto más allá de los típicos cálculos con fórmulas matemáticas. El aprendizaje de dichas consultas fue cuestión de minutos.

#### **4.1.4.6 API REST**

Partiendo de la base de que una API es una forma de describir la manera en que los programas, o sitios Web en este caso, intercambian datos (normalmente en formato JSON o XML), una de las herramientas básicas con las que Global Incubator trabaja en el desarrollo de todos sus proyectos es con las conocidas aplicaciones REST, conocidas como API REST. Una API REST es un servicio que nos provee de funciones que nos dan la capacidad de hacer uso de un servicio web que no es nuestro, dentro de una aplicación propia, de manera segura.[\[20\]](#) Pero en este caso, esta API sí es nuestra.

Este servicio estará limitado a los métodos y funciones de los cual disponga dicha API, de esta forma el creador se asegura de lo que pueden hacer y no los usuarios. Esto es algo muy típico en APIs como Twitter, Facebook o Google. Más adelante hablaré sobre cómo se ha usado dicha herramienta en el proyecto.

### **4.2 Estado inicial del proyecto**

Es importante explicar el estado inicial sobre el que comencé a trabajar en el proyecto, ya que es una forma de ver reflejado el aporte que he realizado en este proyecto, que aún sigue en marcha. Partiendo de la definición de diseño descrita en el [IEEE610.12 – 90] como “el proceso de definir la arquitectura, componentes, interfaces y las otras características de un sistema o componente”, en este punto hablaremos sobre el patrón de diseño escogido y el motivo de esta decisión.

El proyecto comenzó a finales de verano, sobre el mes de Agosto, donde el jefe de la empresa y el encargado de proyectos comenzaron las reuniones con el cliente sobre qué, cómo y cuándo querían realizar el proyecto. Yo entré al proyecto en Septiembre y lo único que tenían hasta la fecha eran las ideas básicas de las funcionalidades que deberían tener la aplicación y un par de bocetos del diseño para ir dando ideas al cliente.

Con los requisitos definidos en el apartado 3.3 y teniendo presente la definición de “diseño”, hemos tratado por separado cada uno de los puntos de dicha definición para poder tomar las decisiones de la manera más justificada, ya que esta parte del diseño nos ayudará a evitar posibles errores futuros causado por un mal diseño. El patrón de diseño elegido ha sido el Modelo-Vista-Controlador, dicho patrón está caracterizado por dividir el software de la aplicación en 3 componentes llamados Modelo, Vista y Controlador, separando la lógica de los datos de la lógica de la vista de la aplicación

Con esto se obtiene una aplicación mucho más modularizada, accesible a la hora de detectar y solucionar posibles errores y con mayor escalabilidad. [21]

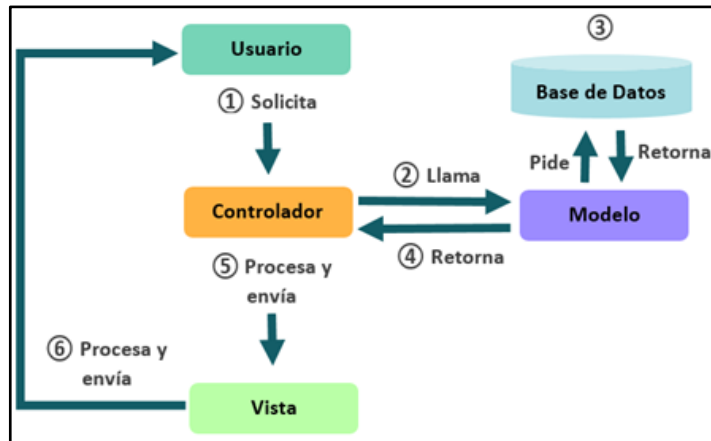


FIGURA 6: MODELO VISTA CONTROLADOR

El **Modelo**, es donde se guarda toda la lógica de la aplicación y abarca desde el acceso a la información en una base de datos hasta el uso que la aplicación haga de esta. Como se aprecia en la figura, a este componente le llega la información del Usuario (1) a través del Controlador (2), que una vez procesada la devuelve (4).

La **Vista**, es el componente que actúa de manera directa con el Usuario. Aquí es donde se encuentra el código HTML, hojas de estilo CSS y archivos Javascript. Cualquier cosa que el Usuario pueda ver, será guardada aquí.

El **Controlador**, es el componente encargado de conectar el Modelo con la Vista. Para ello recoge las acciones que el Usuario solicita en la Vista (1), llama al Modelo (2) que se encarga de obtener los datos (3) y devolver los datos (4) para que el Controlador los procese y envíe de nuevo a la Vista (5).

#### 4.2.1 Diseño del Controlador

Como comenté, el controlador será el encargado de interpretar y dar sentido a las acciones que el usuario realice, es decir, cuando se realice algún cambio en la vista o en el modelo comenzará a actuar. Sin embargo, no vamos a manipular directamente los datos que introduzca el usuario ni a mostrar ninguna salida, sino enlazar el modelo con la vista para implementar las necesidades del desarrollo.

Para este apartado se ha usado el concepto de SPA con el fin de tener varias vistas en lugar de varias pantallas y dar una experiencia más fluida al usuario permitiendo que todas las pantallas se muestren en la misma página sin necesidad de recargar el navegador y sin tener una dirección URL fija. Al tener la aplicación como SPA todos los códigos HTML, CSS y JS están cargados previamente y los recursos necesarios se cargan de forma dinámica

como respuesta a las acciones que realice el usuario, por lo que cuando el usuario interactúe con la SPA, la URL de la barra de direcciones irá cambiando también, pero la clave de esto es que aunque cambie la URL la página no se recargará nunca y mejorará así la velocidad por el renderizado de la web, que al solo cambiar partes más pequeñas dará una experiencia más fluida para el usuario. Este funcionamiento se explicará con más en detalle en el desarrollo de la Vista.

También hay que destacar que el propio navegador mantiene el historial de las pantallas por las que el usuario ha ido navegando, por lo que mediante los botones de “atrás” y “adelante” el usuario podrá usar el historial como si se tratase de una página normal.

Entre los beneficios del uso de SPA y su interacción con el Modelo tenemos la rapidez de las transmisiones de datos y las comunicaciones fluidas entre cliente y servidor, debido al poco peso de los datos. Se comunicará con el Modelo a través del uso de una API REST que obtendrá los datos “en crudo” desde el servidor y los devolverá hacia el cliente en lenguaje JSON, ya que es ligero y tiene soporte con la totalidad de lenguajes usados en la Web. Esto nos da como resultado una aplicación que se comporta como una aplicación como una aplicación de escritorio, respondiendo velozmente al usuario y creando una experiencia muy agradable.

#### 4.2.2 Diseño de la Vista

Una vez que el cliente aceptó las maquetas propuestas por el equipo de Diseño, se empezó a desarrollar el diseño de la Vista, que como se explicó en apartados anteriores se realizó a través del uso de FreeApps, usando en su gran parte lenguaje HTML5-CSS3, jQuery y JavaScript. Para la realización de la interfaz usé una serie de librerías como las que se muestran en la figura 9

```
<link href='[[system:wwwroot]]/app/[[freeappid:92845#attribute:hash]]/?t=5' rel='stylesheet'> /* Call freeApp cms.css */
<link href='https://fonts.googleapis.com/css?family=Raleway' rel='stylesheet'> /* font-family */
<script src='https://maps.googleapis.com/maps/api/js?key=AIzaSyAEZVjFICPnC8pLFX0s-1sUNHw103hYeA&sensor=true&libraries=places'></script> /* Google api */
<link href='https://fonts.googleapis.com/css?family=Noto+Sans' rel='stylesheet'> /* font family */

<script src='//cdnjs.cloudflare.com/ajax/libs/jquery.matchHeight/0.7.0/jquery.matchHeight-min.js'></script> /* Library for function matchHeight */
<script src='https://cdnjs.cloudflare.com/ajax/libs/jquery-locationpicker/0.1.12/locationpicker.jquery.js'></script> /* Library for locationpicker */
<script src='https://cdnjs.cloudflare.com/ajax/libs/jquery-circle-progress/1.2.2/circle-progress.js'></script> /* Library for circle chart */
<meta name='viewport' content='width=device-width, initial-scale=1.0, user-scalable=no' />
<script>var show_onboarding = true;</script> /* show onboarding guide */
```

FIGURA 7: LIBRERÍAS USADAS

Hay muchas más librerías que he usado como jQuery, Bootstrap, Modernizr, Normalize, etc. que no aparecen ahí ya que la propia herramienta de la empresa las tiene precargadas para ser usadas desde cualquier freeApp sin la necesidad de volver a incluirlas. Y en caso de querer que una prevalezca sobre la otra, se vuelve a llamar en el <head> de la freeApp deseada, es importante respetar el orden en que se llaman.

Una de las cosas que más me facilitó el desarrollo de la interfaz de la aplicación fue el uso del framework Bootstrap, ya que gracias al sistema de rejillas pude realizar la distribución de los elementos como el diseño del cliente. Un ejemplo de ello es la figura 10.

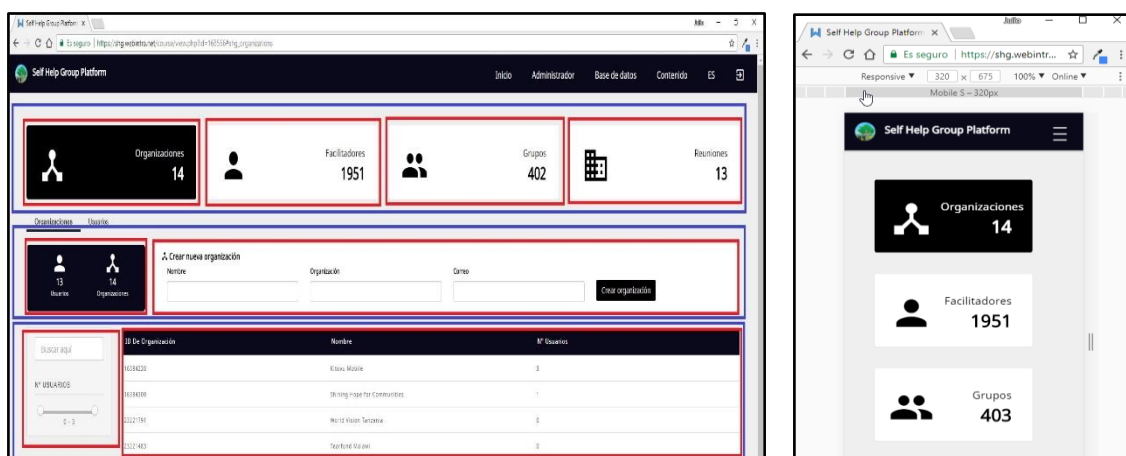


FIGURA 8: APLICACIÓN DE BOOTSTRAP

Bootstrap trabaja principalmente con filas y columnas para colocar los elementos, en la figura podemos ver la interfaz del Admin que, en este caso, dispone de 3 filas (color azul) y una serie de columnas dentro de ellas (color rojo). Como explicamos ya, en Bootstrap cada fila puede contener un máximo de 12 “espacios”, en este caso en la primera fila tenemos 4 columnas de 3 espacios cada una, en la segunda y tercera fila 2 columnas de 2 y 10 espacios respectivamente.

```
<div class="col-xs-12 col-sm-3 box-indicator">
  <section class="panel paneldashboard">
    <div class="box-indicator-icon col-xs-4">
      <i class="material-icons panel-icon" style="color:black;">device_hub</i>
    </div>
    <div class="col-xs-8 panel-data">
      <div class="box-indicator-txt">
        <span>Organizaciones</span>
      </div>
      <span class="box-indicator-value" style="">14</span>
    </div>
  </section>
</div>
```

FIGURA 9: CÓDIGO DE APLICACIÓN DE BOOTSTRAP

Con “col-xs-12 col-sm-3” indico a Bootstrap que para los tamaños ‘xs’, cuando la pantalla sea menor a 768px, el elemento ocupará los 12 espacios de la fila. En cambio, para tamaños ‘sm’ y superiores a 768px, ocupará 3. Aunque la aplicación es desktop first, también se ha hecho pensando en móvil. También en la figura 11 se puede ver cómo se coloca el icono, dándole 4 espacios, y 8 al texto respectivamente. Aunque Bootstrap ya nos facilita por defecto



un estilo para los elementos que vamos situando por nuestro sitio web, he creado una hoja de estilo CSS adaptada a las necesidades del cliente. Dicha hoja de estilo es un fichero aparte, debido a su larga extensión, y ha sido referenciada en cada uno de los ficheros HTML de la aplicación para proporcionar estilo a cada uno de ellos. Aunque el CSS parezca una parte fácil, ha sido necesario consultar foros y tutoriales para aprender acerca del uso de los tipos de selectores css y el uso de jqueries para adaptar cada vista a dispositivos de otro tamaño.

jQuery y JS han jugado un papel importante en la vista ya que me ha permitido tanto la interacción con los distintos elementos del DOM de manera fácil e intuitiva como la creación de funciones para implementar la lógica más “externa”. Un ejemplo de ello se muestra en la figura 12.

```

84- function shg_sendFacilitator(){
85-     var wrapper = $('.form-group');
86-
87-     var btn = wrapper.find('.btn-primary');
88-
89-     var data = {
90-         "id"           : "",
91-         "name"         : wrapper.find('input[id=usr_new_facilitator]').val(),
92-         "address"      : wrapper.find('input[id=address_new_facilitator]').val(),
93-         "location"     : wrapper.find('input[id=latitude_new_facilitator]').val() + ', '
94-         + wrapper.find('input[id=longitude_new_facilitator]').val(),
95-         "organization" : wrapper.find('input[id=org_new_facilitator]').val(),
96-         "date_added"   : wrapper.find('input[id=date_new_facilitator]').val(),
97-         "email"        : wrapper.find('input[id=email_new_facilitator]').val(),
98-         "phone"        : wrapper.find('input[id=phone_new_facilitator]').val()
99-     }
100-
101-     btn.addLoader(function(){
102-         getServiceResponse("services/aml_views/view/public_view.php",{
103-             id : "[[freeapp:89462:attribute:hash]]",
104-             h : true,
105-             clean: true,
106-             params : $.toJSON(data)
107-         },function(response){
108-             //Cerrar popup
109-             alert("Facilitator has been created successfully");
110-             btn.removeLoader();
111-         }, false, function(error){
112-             btn.removeLoader();
113-             alert(error);
114-         })
115-     })
116- }

```

FIGURA 10: USO DE JQUERY Y JS

En dicho fragmento de código, declaro mediante JS una función para enviar los datos de un nuevo facilitador y, mediante jQuery, accedemos a la clase del formulario de registro y la guardamos en la variable “wrapper” (ln 85) para posteriormente buscar y guardar en “btn” la clase asociada al botón del envío de datos del formulario (ln 87). La variable “data” es de tipo objeto y mediante jQuery accede al DOM y recoge el valor de los datos introducidos por el usuario en el formulario (ln 89). En la ln 100, se añade un “loader” al botón de envío para que indique que se están procesando los datos. Como parámetro llama a la función “getServiceResponse” la cual contiene como primer argumento la URL para conectar con la el servicio Web encargado de guardar los datos, como segundo un objeto con el id y hash de la freeApp que procesará los datos (ln 103), serán enviados en formato JSON (ln 106) y como tercero y último, realiza un callback (ln 107) para devolver si dicho proceso fue correcto o alertar de lo contrario.

Por último, y aunque su aplicación no tiene que ver con el diseño de las interfaces, quiero comentar el uso de la tecnología AJAX en este apartado, ya que es donde más funcionalidad tiene. AJAX se ha aplicado en todos los cambios de vistas Web y en los buscadores. Gracias a ello hemos permitido que el sitio Web tenga dinamismo, siendo capaz de cambiar el contenido de la misma sin la necesidad de que el servidor tenga que refrescar la página. Es una forma de incorporar lógica en el lado del cliente para ayudar a mejorar la funcionalidad de la página para que no esté toda en el lado del servidor. La figura 14 muestra como a través de esta tecnología al cambiar de vista simplemente basta con cambiar el “hash” de la URL de esa vista y mediante AJAX se cambia por “hash” de la nueva vista seleccionada.

```
329 - function manage_shg_hash(){
330     var hash = location.hash;
331     if(hash == "")
332     {
333         return false;
334     }
335     hash = hash.substr(1);
336
337     var object = $('[data-content-hash="'+ hash + '"]:first');
338     $(".panel_option_button.selected").removeClass("selected");
339     object.addClass("selected");
340     var freeappId = object.attr("data-content-id");
341
342     loadContent(freeappId);
343 }

345 - function loadContent(id){
346     connector('services/aml_views/view/public_view.php',{
347         id : id,
348         h : true,
349         clean: true
350     }, function(response){
351         $('main').empty().append(response);
352     });
353 }
```

FIGURA 11: FUNCIONES AJAX

Como podemos ver en la figura 13, este funcionamiento es gestionado mediante dos funciones. Por un lado con “manage\_shg\_hash” usamos “location.hash” para obtener la parte que acompaña al # de la URL actual (incluyendo el #) y con “hash.substr(1)” extraemos el primer carácter de esa cadena. Una vez tenemos la cadena sin el carácter #, usamos el “hash” para encontrar el div que tiene esa clase y mediante selectores de jQuery removemos la clase “selected” actual y añadimos la clase “selected” para que los css se pongan como seleccionados.

Finalmente se guarda en “freeappId” el valor hash almacenado en “data-content-id” y se pasa como argumento para la llamada de la función “loadContent”. En esta función, se usa la función connector que realiza una llamada a la freeApp con la vista que va a mostrarse, y la pinta en \$('main'). Connector se encarga de realizar la funcionalidad de AJAX a la URL que reciba por los parámetros, como tercer parámetro realiza un callback que se ejecuta si todo fue correcto.

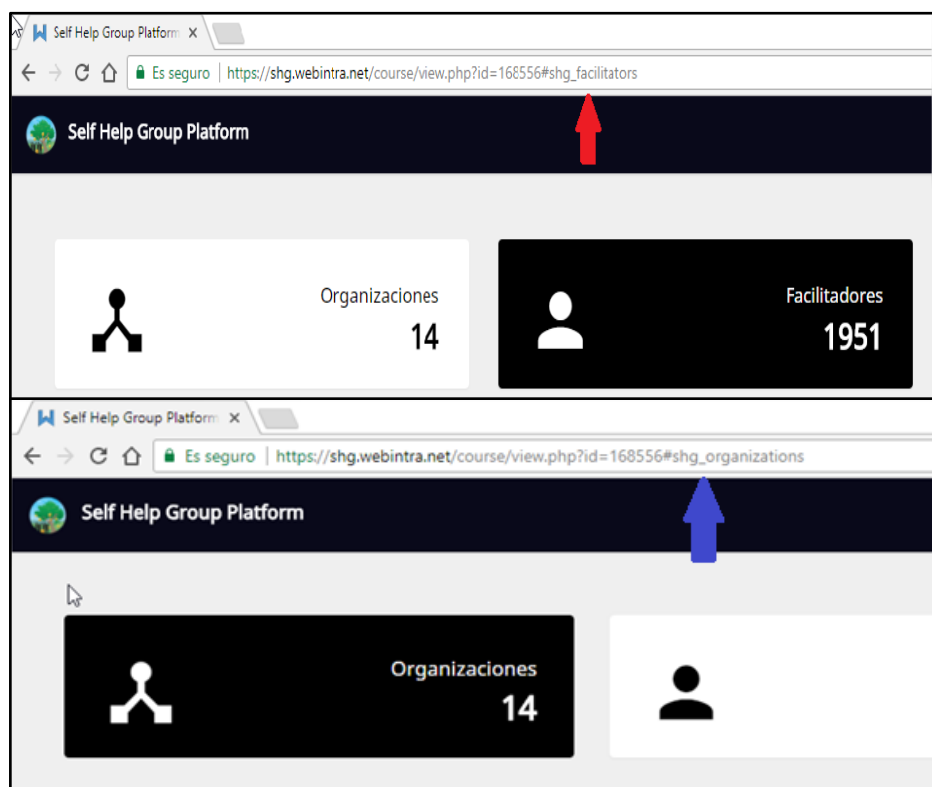


FIGURA 12: EJEMPLO 1 DE AJAX

En el caso de los buscadores, el funcionamiento es el siguiente: Cuando el usuario quiera buscar, en este caso, un usuario miembro de una organización deberá rellenar sus datos, pulsar sobre un botón y se le mostrará una tabla con los resultados. Con el objetivo de evitar recargar la vista para realizar dicha búsqueda, y para facilitar la experiencia al usuario tanto por fluidez como por ayuda en caso de que no se sepa los datos completos del usuario se usa AJAX, de modo que la aplicación irá ejecutando queries en la base de datos a medida que el usuario vaya introduciendo caracteres en el buscador, y se irán mostrando las coincidencias en la tabla

La figura 15 muestra dicho funcionamiento, de esta manera al escribir simplemente la letra “j”, tenemos todas las coincidencias de los usuarios con dicha letra.

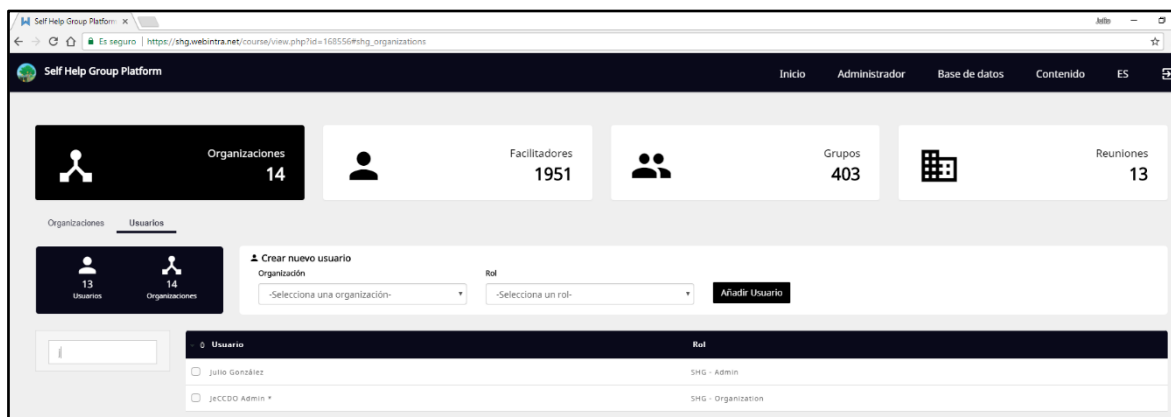


FIGURA 13: EJEMPLO 2 DE AJAX

### 4.2.3 Diseño del Modelo

Como comenté anteriormente, el Modelo guardará toda la lógica de la aplicación y en este caso, será el encargado del manejo del acceso a la información de la base de datos.

Para esta aplicación necesitaremos guardar y acceder a la información que el usuario introduzca tanto desde la aplicación Web como desde la aplicación móvil del cliente. Dado que la aplicación móvil trabaja con una base de datos local pensada para funcionar offline, debido a la baja conexión a Internet de los lugares donde se usa, se ha usado API REST para que cada vez que dicha base local se conecte a Internet se sincronice toda la información que tenga almacenada con la base de datos de la aplicación Web mediante el uso de los métodos API REST.

API REST es un tipo de arquitectura para desarrollo Web que se basa en el estándar HTTP, por lo tanto para poder hacer uso de la API hay que llamarla con peticiones HTTP en las que una URI representa el recurso (con identificador único) al que queremos acceder. La URL donde se encuentra la definición de todos los métodos de la API es: <https://webintra.net/course/view.php?id=11970#api>.

Las APIs usan principalmente 4 métodos HTTP estándar como GET (leer y consultar), POST (crear), PUT (editar) y DELETE (eliminar), por lo que en función del tipo de métodos de los que disponga nuestra API podremos hacer una serie de acciones con los recursos. Para esta aplicación se han usado los siguientes:

- POST para crear nuevos recursos
  - La URL permanecerá “abierta” ya que el recurso aún no existe y no tiene un ID asignado

- URL: POST shg.webintra.net/api/SHG/facilitator
- Códigos de estado de respuesta del servidor
  - 400, si la petición es incorrecta (por ejemplo, falta algún campo o es un valor inválido).
  - 500, si el error se produce en el lado del servidor (por ejemplo, fallo en la base de datos).
  - 201, si el recurso se creó correctamente.
- GET para obtener recursos
  - URL GET  
shg.webintra.net/api/SHG/facilitators?sort={0}&limit={0}&page={0}&name={0}&order={string}
  - Códigos de estado de respuesta del servidor
    - 200 OK, si se obtuvo correctamente.
    - 404 NOT FOUND, si no se encontró el recurso
- DELETE para eliminar recursos
  - URL DELETE shg.webintra.net/api/SHG/group?id={string}
  - Tras ejecutar el DELETE con éxito, las siguientes peticiones GET a la URL del recurso deberían devolver 404.

Hemos usado API REST porque nos ha permitido una serie de ventajas como:

- Interfaz sencilla y escalable, es decir, basada en recursos (como por ejemplo el recurso Facilitador (id, nombre, organización, etc.)). Mientras la interfaz no cambie, podremos cambiar el cliente o el servidor sin problemas.
- Servidor transparente para el cliente, nos manda los datos vía JSON pero lo que este contenga en su interior (BBDD) es ajeno para el cliente.
- Representación múltiple del recurso, permitiendo a los recursos ser representados en distintos formatos diferentes a aquel en el que están almacenados. Por ejemplo, puede estar almacenado en formato XML y ser representado en JSON (como es el caso).
- Uso de mensajes descriptivos, al usar el protocolo HTTP la semántica se vuelve más comprensible (HTTP Verbs, HTTP status codes, etc)
- Separación cliente-servidor, se unen mediante la interfaz y mientras que esta no cambie, podremos cambiar el cliente o el servidor sin problemas.

- Comunicaciones sin estado, esto es porque el servidor trata cada petición como algo totalmente independiente y la comunicación siempre devuelve un estado final, es decir, siempre devuelve el recurso completo sin opción a pedir al servidor en la siguiente transición que nos envíe la otra parte del recurso ya que no sabría a qué nos referimos. Con esto conseguimos proteger al cliente de los cambios que se puedan producir en el servidor, ya que no tiene que preocuparse sobre con quien se comunica.

La figura 16 muestra parte de la estructura de la API SHG (recuadro rojo). En ella podemos ver la dirección de acceso (recuadro azul), el listado de todos los métodos que contiene y el método POST/facilitator (recuadro marrón) desplegado con los parámetros necesarios para invocarlo (recuadro naranja) y la muestra de la URL a usar (recuadro verde).

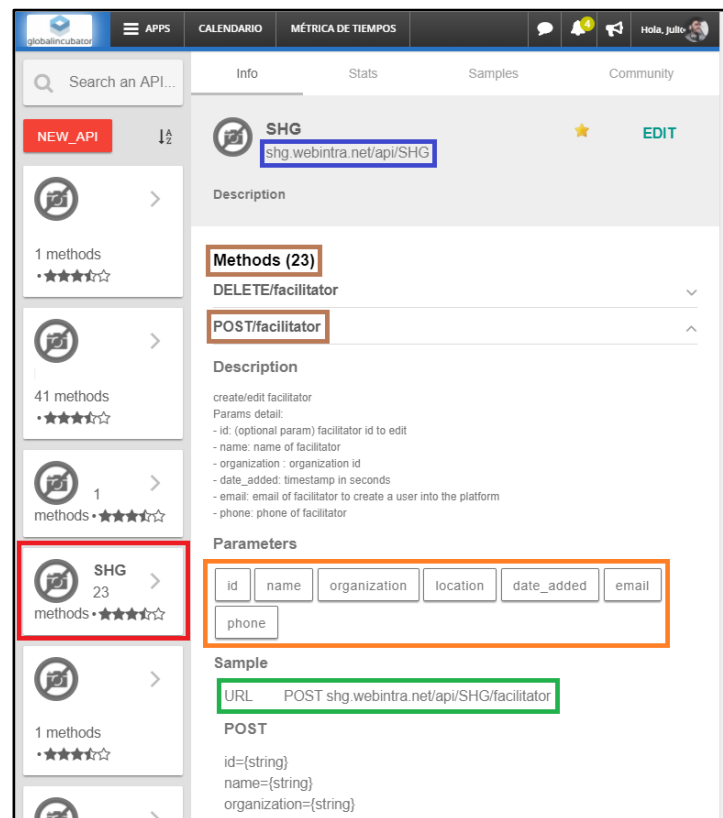


FIGURA 14: API SHG

## 5 Integración, pruebas y resultados

Hay que decir que al ser una aplicación real, aún quedan pruebas por realizar y funcionalidades que podrían variar ligeramente en el producto final y normalmente esta sería la última etapa del ciclo de vida del desarrollo de un proyecto pero en nuestro caso, al haber optado por un ciclo de vida de software iterativo e incremental, hemos ido realizando pruebas e integraciones al final de cada ciclo

Las pruebas que se han realizado tienen como fin buscar situaciones o partes de la aplicación donde se crea que el funcionamiento de esta puede ser diferente al esperado, como por ejemplo:

- Entradas de datos a la aplicación
- Tiempo de respuesta en cargar contenidos
- Funciona en cualquier dispositivo y tamaño (pc, Tablet y móvil)
- Respuesta antes situaciones inesperadas

Las pruebas realizadas en cada ciclo se han dividido en pruebas funcionales y no funcionales. Para la parte que he realizado me he enfocado en las de carácter funcional, ya que las de carácter no funcional se realizarán principalmente en la parte final del proyecto.

Dentro de las pruebas funcionales he realizado pruebas unitarias, pruebas de integración y pruebas de validación, siguiendo un orden lógico desde las más básicas hasta los niveles más altos de abstracción.

El orden de realización de las pruebas ha sido el siguiente:

#### Funcionales

##### *1. Pruebas unitarias*

Realizadas en primer lugar para que los posibles fallos que ocurran en las pruebas de integración no se deban a fallos de algún módulo.

Se han focalizado en ejecutar cada módulo de la aplicación de forma individual, de los más fáciles a los más complejos, para comprobar su funcionamiento por separado mediante el diseño de casos de pruebas sin tener en cuenta otras funcionalidades. Se han considerado aspectos como: rutinas de excepción, rutinas de error, manejo de parámetros, etc.

##### *2. Pruebas de integración*

Realizadas para identificar errores que se hayan producido tras combinar los módulos probados unitariamente.

Por cada caso de prueba ejecutado se ha comparado el resultado esperado con el resultado obtenido, usando técnicas como top-down para probar que los módulos de nivel superior llamen a los de nivel inferior de forma correcta y down-town para el caso contrario.

##### *3. Pruebas de validación*

Por último, estas pruebas son las encargadas de comprobar que la funcionalidad de la aplicación, hasta ese momento, es la especificada en el análisis de requisitos.

Se han dividido en pruebas Alfa y pruebas Beta, ya que son pruebas utilizadas para detectar errores que sólo el usuario final (el cliente en este caso) es capaz de detectar.

- a. Pruebas Alfa, se han realizado dentro de la propia empresa con distintos compañeros de trabajo para detectar problemas que los miembros del proyecto no hemos visto
- b. Pruebas Beta, se han realizado únicamente por el cliente, sin presencia de ningún miembro del equipo con el objetivo de que el cliente tenga total libertad de probar la funcionalidad y reportar cualquier error observado en ese ciclo.

A continuación, se muestra una tabla con algunas las pruebas de validación realizadas hasta el estado actual del proyecto en función a los requisitos funcionales descritos en el Apartado 3.3

Prueba	Opciones	Descripción	Salida esperada
<b>LogIn</b>	Correcto	El usuario accede a la aplicación con su usuario y contraseña	El usuario accede a la aplicación.
	Incorrecto	El usuario accede a la aplicación con un usuario y contraseña erróneos.	El sistema muestra un mensaje de error de acceso al usuario.
<b>Cambio de idioma</b>	Español/ Inglés/ Francés	El usuario pulsa sobre el botón de idiomas situado en la esquina superior derecha, elige el nuevo idioma y pulsa “Aceptar” para confirmar su acción.	El sistema refresca la página actual y cambia el idioma.
<b>Ver info y editar coordenadas de un grupo</b>	Cambiar latitud y/o longitud	El usuario, una vez situado en la pestaña “Groups”, selecciona la columna de las coordenadas del grupo que desea visualizar o cambiar y pulsa “save” para guardar la nueva ubicación.	El sistema guarda la acción y muestra la nueva localización del grupo.
	Combinación de caracteres válida.	El usuario, una vez situado en la pestaña “Groups”, introduce en el buscador cualquier combinación válida de caracteres por el que quiera buscar. Por ejemplo, “pam”	El sistema le muestra las coincidencias encontradas.



<b>Buscar un grupo</b>	Combinación de caracteres inválida.	El usuario, una vez situado en la pestaña “Groups”, introduce en el buscador cualquier combinación inválida de caracteres por el que quiera buscar. Por ejemplo, ‘3jkgp’.	El sistema no muestra nada.
<b>Buscar un facilitador</b>	Rol Organization	El usuario, una vez situado en la pestaña “Facilitators” ve los facilitadores de su organización y los filtra por: Cantidad de personas de sus grupos, Fecha de adicción y Fecha de actualización.	El sistema le devuelve los datos en función del filtro escogido
	Rol SHG Admin	El usuario, una vez situado en la pestaña “Facilitators” ve los facilitadores de su organización y los filtra por: Organización, Cantidad de personas de sus grupos y datos inconsistentes.	El sistema le devuelve los datos en función del filtro escogido
<b>Crear nuevo usuario</b>	Rol SHG Admin / Entradas completas	El Admin, una vez situado sobre la pestaña “Administrator” > ”Users”, rellena los campos “Organization” y “Role” y pulsa sobre “Add User”	El sistema informa de la creación correcta del usuario y lo añade al listado
	Rol SHG Admin / Entradas incompletas	El Admin, una vez situado sobre la pestaña “Administrator” > ”Users”, no rellena alguno de los campos “Organization” y “Role” y pulsa sobre “Add User”	El sistema informa que debe rellenar todos los campos.
<b>Crear nueva organización</b>	Rol SHG Admin / Entradas completas	El Admin, una vez situado sobre la pestaña “Administrator” > ”Organizations”, rellena los campos “Name”, “Organization” y “Mail” y pulsa sobre “Create Organization”	El sistema informa de la creación correcta de la organización y la añade al listado
	Rol SHG Admin / Entradas incompletas	El Admin, una vez situado sobre la pestaña “Administrator” > ”Organizations”, y no rellena algunos de los campos “Name”, “Organization” y “Mail” y pulsa sobre “Create Organization”	El sistema informa que debe rellenar todos los campos.

<b>Consultar base de datos</b>	Rol SHG Admin	El Admin pulsa sobre la pestaña “Database” y con el desplegable selecciona por qué campo quiere filtrar.	El sistema le muestra los resultados de acorde al filtro aplicado
--	---------------------	--	---

Todas las pruebas que aquí se muestran han sido superadas satisfactoriamente y la salida esperada encaja con los requisitos descritos anteriormente.

En el Anexo D pueden verse algunas capturas de las pruebas realizadas.

## 6 Conclusiones y trabajo futuro

---

### 6.1 Conclusiones

En este apartado comentaré las sensaciones finales que he tenido tras la elaboración de dicho proyecto, tanto a nivel personal como técnico.

A nivel técnico, estoy bastante contento ya que he sabido poner en práctica muchos de los conceptos aprendidos en asignaturas como Proyecto de Análisis y Diseño de Software, Sistemas Informáticos, Estructuras de Datos o Ingeniería del Software, y he visto de primera mano cómo es el trato con un cliente y cómo dista la teoría de la práctica en la vida real durante el desarrollo de un proyecto, y lo que en un principio parece fácil, realmente no lo es.

Ha habido momentos que han sido más difícil a la hora de tratar con el cliente ya que, debido al idioma y a la distancia, la comunicación ha sido en inglés y principalmente a través de Skype y WhatsApp, y a veces es más difícil hacerse entender así.

Pero siendo sincero, me ha gustado el “reto” de tener que trabajar en inglés, ya que me ha hecho aprender conceptos técnicos y sé que en el futuro es algo que me vendrá genial, ya que mi intención es trabajar fuera de España.

Pero no ha sido todo tan “fácil”, ya que a pesar de todos los conocimientos de las asignaturas mencionadas, he tenido que aprender bastantes cosas nuevas, sobretudo el lenguaje AML, propio de la empresa, el cual ha sido uno de los mayores retos debido a la casi inexistente documentación de la cual disponían. También, respecto al diseño Web, me he dado cuenta de que es mucho más extenso de lo que imaginaba y tiene todo un mundo detrás para aprender a diseñar una buena Web. Tal ha sido el interés que me ha despertado

este tema que compré un par de cursos online sobre desarrollo Web para seguir formándome sobre este tema.

A nivel personal, me ha parecido una gran experiencia el poder ser parte de un proyecto real con objetivos sociales como los comentados, ya que saber que con tu trabajo puedes ayudar a otras personas es algo muy gratificante y sé que de cara al futuro me va a servir el haber realizado una experiencia así.

También he visto que soy capaz de trabajar con más compañeros y formar parte de un proyecto gracias a lo aprendido durante la carrera y también a conceptos que no se practican como el trato con clientes reales. Y como mencioné antes, el haber realizado el feedback con el cliente completamente en inglés.

## ***6.2 Trabajo futuro***

A pesar de haber completado la mayor parte de los requisitos funcionales del proyecto, al tratarse de un proyecto real, aún falta trabajo por delante.

Algunos de los aspectos que se mejorarán son los siguientes:

- Se permitirá al Admin poder eliminar organizaciones y usuarios desde la aplicación, ya que de momento sólo puede hacerse desde el backend por parte del equipo de programadores.
- Cuando se creen nuevos facilitadores, su número de identificación en lugar de ser un número de 15 cifras creado de forma aleatoria, será el IMEI del móvil de cada facilitador.
- El Admin podrá gestionar, en la pestaña CMS, el contenido en la aplicación móvil, añadiendo nuevas unidades, pasos y módulos.
- Se incluirá el swaghili como nuevo idioma
- Se permitirá el envío de correos electrónicos desde la propia aplicación para facilitar la difusión de mensajes de un organizador a sus facilitadores.
- Se incluirán gráficas que muestren el progreso de los ahorros de cada grupo y, a su vez, un sistema que trate de predecir cuáles serán los ahorros estimados en el futuro.

# Referencias

---

- [1] Era Digital  
[https://es.wikipedia.org/wiki/Era\\_de\\_la\\_informaci%C3%B3n](https://es.wikipedia.org/wiki/Era_de_la_informaci%C3%B3n)
- [2] Evelio Martínez, Qué es la Brecha Digital, La Brecha Digital, 1, 22 de Enero de 2008  
<http://www.labrechadigital.org/labrecha/qu-es-la-brecha-digital17.html>
- [3] Alfonso Domínguez Calvente, ¿Era Digital? ¿Y en los países subdesarrollados?, Blog EOI, 1, 16 de Junio de 2012.  
<http://www.eoi.es/blogs/alfonsodominguez/2012/06/16/%C2%BFera-digital-%C2%BFy-en-los-paises-subdesarrollados/>
- [4] JavaScript  
<https://es.wikipedia.org/wiki/JavaScript>
- [5] JavaScript Library Usage  
<https://qph.ec.quoracdn.net/main-qimg-0427609a96427a985411930704885021>
- [6] JQuery  
<https://es.wikipedia.org/wiki/JQuery#Caracter.C3.ADsticas>
- [7] DOM  
[https://es.wikipedia.org/wiki/Document\\_Object\\_Model](https://es.wikipedia.org/wiki/Document_Object_Model)
- [8] HTML  
<https://es.wikipedia.org/wiki/HTML>
- [9] CSS  
[https://es.wikipedia.org/wiki/Hoja\\_de\\_estilos\\_en\\_cascada](https://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada)
- [10] AJAX  
<https://es.wikipedia.org/wiki/AJAX>
- [11] Bootstrap  
[https://es.wikipedia.org/wiki/Bootstrap\\_\(framework\)#Caracter.C3.ADsticas](https://es.wikipedia.org/wiki/Bootstrap_(framework)#Caracter.C3.ADsticas)
- [12] Bootstrap rejillas  
[https://ajgallego.gitbooks.io/bootstrap-3/content/capitulo\\_rejilla.html](https://ajgallego.gitbooks.io/bootstrap-3/content/capitulo_rejilla.html)
- [13] Bootstrap rejillas imagen  
<https://image.slidesharecdn.com/sitiowebbootstrap06-141005175240-conversion-gate02/95/sitio-web-bootstrap-3-10-638.jpg?cb=1412676693>
- [14] SQL

- <https://es.wikipedia.org/wiki/SQL>
- [15] Google Drive  
[https://es.wikipedia.org/wiki/Google\\_Drive](https://es.wikipedia.org/wiki/Google_Drive)
- [16] Trello  
<https://es.wikipedia.org/wiki/Trello>
- [17] WhatsApp  
<https://es.wikipedia.org/wiki/WhatsApp>
- [18] Microsoft Excel  
[https://es.wikipedia.org/wiki/Microsoft\\_Excel](https://es.wikipedia.org/wiki/Microsoft_Excel)
- [19] Microsoft Word  
[https://es.wikipedia.org/wiki/Microsoft\\_Word](https://es.wikipedia.org/wiki/Microsoft_Word)
- [20] Andrearrs, ¿Qué es una API?, hipertextual, 1, 15 de Mayo de 2014  
<https://hipertextual.com/archivo/2014/05/que-es-api/>
- [21] Rodrigo Gómez, Modelo Vista Controlador, Rodrigo Gómez, 1, 11 de Noviembre de 2015  
<http://rodrigogr.com/blog/modelo-vista-controlador/>
- [22] Internet  
<https://es.wikipedia.org/wiki/Internet>
- [23] PHP  
<https://es.wikipedia.org/wiki/PHP>
- [24] Leandro Alegsa, Ciclo de vida software, Alegsa, 1, 6 de Septiembre de 2009  
[http://www.alegsa.com.ar/Dic/ciclo\\_de\\_vida\\_del\\_software.php](http://www.alegsa.com.ar/Dic/ciclo_de_vida_del_software.php)
- [25] Javier Garzas, Ciclos de vida para gestionar un proyecto software: cascada, espiral, iterativo, incremental o ágil, JavierGarzas.com, 1, 3 de Julio de 2013  
<http://www.javiargarzas.com/2013/07/ciclos-de-vida-software.html>
- [26] Ángel Miguel Macas, Diseño de Software, Monografias.com, 2  
<http://www.monografias.com/trabajos73/disen-software/disen-software.shtml>

## Glosario

---

TIC	Tecnologías de la Información y la Comunicación
API	Interfaz de programación de aplicaciones
REST	Transferencia de Estado Representacional
URL	Localizador uniforme de recursos
URI	Identificador uniforme de recursos
SPA	Aplicación de página simple
Startup	Empresa emergente.
Feedback	Intercambio de opciones con un con el cliente.
Dashboards	Panel Web.
Framework	Entorno de trabajo.
Internet	Conjunto descentralizado de redes de comunicación interconectadas.
Base de datos	Colección de información organizada.
Base relacional	Conjunto de tablas que contienen datos provistos en categorías predefinidas.
Clave Primaria	Campo que identifica de manera única a cada entidad.
Clave Externa	Campos que almacenan claves primarias de otras tablas.
JQuery	Framework de JavaScript.
DOM	Document Object Mode.
JavaScript	Lenguaje de programación interpretado.
JSON	JavaScript Object Notation.
PHP	Lenguaje de programación diseñado para uso en Web del lado del servidor.
HTML	HyperText Markup Language.
CSS	Cascading Stylesheet.
Navegador Web	Aplicación que permite el acceso a la Web.
Página Web	Información electrónica que puede ser accedida mediante un navegador Web.
Contenido Web	Conjunto de elementos que formar parte de la página Web.
Responsive Web	

Design	Filosofía de diseño y desarrollo Web cuyo objetivo es adaptar la apariencia de las páginas web.
Bootstrap	Framework para el diseño de sitios Web.
Desktop first	Tipo de diseño web centrado en formatos de pantalla a partir de 768px
Normalize	Código CSS que resetea las propiedades CSS de un sitio Web.
Modernizr	Código CSS para que una Web funcione con todas sus capacidades en cualquier navegador.
SQL	Lenguaje para gestionar bases de datos relacionales.
Sistema operativo	Conjunto de programas de un sistema informático que gestiona los recursos del mismo.
Google Docs	Programa gratuito basado en Web para crear documentos en línea con la posibilidad de colaborar en grupo.
Sprint	Período en el cual se lleva a cabo el trabajo propuesto.
Microsoft Office	Recopilación de aplicaciones informáticas utilizadas en oficinas, para realizar diferentes funciones sobre archivos y documentos.
Microsoft Word	Programa informático del paquete Microsoft Office informático orientado al procesamiento de textos.
Microsoft Excel	Programa informático del paquete Microsoft Office orientado al uso y manejo de hojas de cálculo.
WhatsApp	Aplicación de mensajería instantánea para teléfonos inteligentes.
AML	Pseudo-lenguaje de programación basado en PHP y JavaScript.
Apps	Módulos independientes para el desarrollo de aplicaciones.
QApps	Tipo de App para la creación de formularios.
FreeApps	Tipo de App para la creación de vistas Web.
Heurística de Nielsen	Serie de pasos a seguir para crear sistemas que sean amigables para el usuario.





## Anexos

---

### *A Requisitos de la aplicación*

#### **Requisitos funcionales**

Los requisitos funcionales son declaraciones de los distintos servicios que debe proporcionar un sistema, especificando la manera en la que dicho sistema debe reaccionar a las acciones del usuario. A continuación se muestran en función al rol de entrada en la aplicación.

#### *Comunes a Facilitador, Organización y Admin*

##### [RF 1] – Iniciar sesión en la aplicación (Log-In)

El usuario debe poder iniciar sesión en la aplicación utilizando su nombre de usuario (dirección de correo electrónico) y su contraseña.

##### *Entrada:*

El usuario accede a la ventana o página de Log-In de la aplicación e introduce su nombre de usuario y su contraseña, después pulsa en el botón “Entrar”.

##### *Sistema:*

El sistema verifica que ese usuario está registrado en la base de datos de la aplicación y verifica también que su contraseña coincide.

##### *Salida:*

Si todo es correcto muestra al usuario la pantalla principal de la aplicación con la funcionalidad correspondiente según el tipo de usuario que sea. Si ha habido algún error durante la verificación de datos se lo muestra al usuario por pantalla indicando sus causas e invitándole a que vuelva a intentarlo.

##### [RF 2] – Cierre de sesión (Log-Out)

El usuario podrá cerrar sesión en la aplicación tras pulsar el botón y confirmar su salida.

##### *Entrada:*

El usuario pulsa sobre el botón de la esquina superior derecha y pulsa “Aceptar” para confirmar su acción.

*Sistema:*

Pregunta al usuario si realmente desea salir de la aplicación.

*Salida:*

El usuario sale de la aplicación.

#### [RF 3] – Cambio de idioma

El usuario podrá cambiar el idioma de la aplicación en cualquier momento.

*Entrada:*

El usuario pulsa sobre el botón de idiomas situado en la esquina superior derecha, elige el nuevo idioma y pulsa “Aceptar” para confirmar su acción.

*Sistema:*

Pregunta al usuario si realmente desea salir de la aplicación. Si la respuesta afirmativa el Sistema refresca la página para realizar los cambios.

*Salida:*

El Sistema muestra el mismo contenido que habia anteriormente pero en el nuevo idioma seleccionado.

#### [RF 4] – Guía de bienvenida

El usuario podrá ver una guía rápida de uso de la aplicación.

*Entrada:*

El usuario, tras iniciar sesión, elige si acceder a la guía rápida o si ir directamente a la aplicación.

*Sistema:*

Cada vez que el usuario inicie sesión o refresque la página en la que se encuentra el Sistema cargará una guía rápida de bienvenida.

*Salida:*

El Sistema muestra una guía rápida personalizada en función al rol con el que se inició la sesión actual.

#### [RF 5] – Consultar información de grupos

El usuario actual podrá consultar la información asociada a cada grupo.

*Entrada:*

El usuario pulsa sobre la pestaña “Home” > “Groups” o directamente sobre la pestaña “Groups” (disponible sólo para rol de Organización y Facilitador) para poder ver los grupos, pudiendo ser filtrados por: total savings, total loans, kind, nºmales, nºfemales, activity e inconsistent data.

*Sistema:*

El Sistema ejecuta una consulta a la base de datos para obtener un listado de todos los grupos asociados a ese usuario. Si se establece algún filtro, la consulta tendrá en cuenta los parámetros del filtro establecidos para ejecutar la consulta.

*Salida:*

El Sistema, por defecto, muestra una tabla con toda la información relativa a un grupo o de la correspondiente a la aplicación del filtro que se utilizó.

[RF 6] – Ver un grupo en detalle

El usuario actual, tras acceder a la información de grupos, podrá ver información más detallada de cada grupo.

*Entrada:*

El usuario pulsa sobre la pestaña “Home” > “Groups” > “View” o directamente sobre la pestaña “Groups” > “View” (disponible sólo para rol de Organización y Facilitador).

*Sistema:*

El Sistema ejecuta una consulta a la base de datos para obtener la información detallada de ese grupo.

*Salida:*

El Sistema muestra los detalles de ese grupo, la geolocalización a través de un mapa y el progreso de reuniones de ese grupo.

[RF 7] – Editar las coordenadas de un grupo

El usuario actual, tras acceder a la información de grupos, podrá editar las coordenadas de cada grupo.

*Entrada:*

El usuario pulsa sobre la pestaña “Home” > “Groups” o directamente sobre la pestaña “Groups” > “View” (disponible sólo para rol de Organización y Facilitador) y, sobre la tabla mostrada, pulsa “View”.

En lugar de pulsar sobre “View” también podrá pulsar sobre la columna “Coords” del grupo que quiera editar.

Una vez ahí, podrá cambiar la longitud y/o latitud de ese grupo.

*Sistema:*

El Sistema ejecuta una llamada para obtener la vista con la información detallada de ese grupo.

*Salida:*

El Sistema muestra la nueva geolocalización de ese grupo a través de un mapa.

[RF 8] – Consultar grupos de un facilitador

El usuario actual, podrá ver qué grupos tiene asociado un facilitador.

*Entrada:*

El usuario pulsa sobre la pestaña “Home” > “Groups” o directamente sobre la pestaña “Groups” (disponible sólo para rol de Organización y Facilitador) y, sobre la tabla mostrada, elige el ID del facilitador que quiere consultar.

Para los roles de Admin y Organización, el usuario también podrá acceder pulsando el botón “Facilitators” y, sobre la tabla mostrada, pulsar en “View groups”.

*Sistema:*

El Sistema ejecuta una llamada para obtener la vista con la información de los grupos de ese facilitador.

*Salida:*

El Sistema muestra, en forma de tabla, los grupos asociados a ese facilitador.

[RF 9] – Consultar información de reuniones

El usuario podrá consultar la información asociada a cada una de las reuniones realizadas.

*Entrada:*

El usuario pulsa sobre la pestaña “Home” > “Meetings” para poder ver reuniones realizadas, pudiendo ser filtrados por los puntos marcados en el checklist.

*Sistema:*

El Sistema ejecuta una consulta a la base de datos para obtener una tabla con todas las reuniones realizadas por ese usuario. Si se establece algún filtro, la consulta tendrá en cuenta los parámetros del filtro establecidos para ejecutar la consulta.

*Salida:*

El Sistema, por defecto, muestra una tabla con toda la información de cada reunión o de la correspondiente a la aplicación del filtro que se utilizó.

***Específicos para el Subsistema Organización***

[RF 10] – Consultar información de los facilitadores

El usuario podrá consultar la información asociada a cada una de los facilitadores.

*Entrada:*

El usuario pulsa sobre la pestaña “Home” > “Facilitators” o directamente sobre la pestaña “Facilitators” para poder ver los facilitadores de su organización, pudiendo ser filtrados por: Cantidad de personas de sus grupos, Fecha de adicción y Fecha de actualización.

*Sistema:*

El Sistema ejecuta una consulta a la base de datos para obtener una tabla con los grupos que tiene esa Organización. Si se establece algún filtro, la consulta tendrá en cuenta los parámetros del filtro establecidos para ejecutar la consulta.

*Salida:*

El Sistema, por defecto, muestra una tabla con toda la información de cada grupo o de la correspondiente a la aplicación del filtro que se utilizó.

***Específicos para el Subsistema Administrador***

[RF 11] – Consultar información de los facilitadores

El Admin podrá consultar la información asociada a cada una de los facilitadores.

*Entrada:*

El Admin pulsa sobre la pestaña “Home” > “Facilitators” para poder ver los facilitadores de su organización, pudiendo ser filtrados por: Nombre de la Organización, Cantidad de personas de sus grupos y Datos inconsistentes.

*Sistema:*

El Sistema ejecuta una consulta a la base de datos para obtener una tabla con toda la información relativa a cada uno de los facilitadores. Si se establece algún filtro, la consulta tendrá en cuenta los parámetros del filtro establecidos para ejecutar la consulta.

*Salida:*

El Sistema, por defecto, muestra una tabla con toda la información de cada facilitador o de la correspondiente a la aplicación del filtro si se utilizó.

[RF 12] – Consultar información de las organizaciones

El Admin podrá consultar la información asociada a cada una de las organizaciones.

*Entrada:*

El Admin pulsa sobre la pestaña “Home” > “Organizations” o directamente sobre la pestaña “Administrator” para poder ver todas las organizaciones, pudiendo ser filtrados por: n° de usuarios.

*Sistema:*

El Sistema ejecuta una consulta a la base de datos para obtener una tabla de todas las organizaciones existentes.

*Salida:*

El Sistema, por defecto, muestra una tabla con toda la información de cada organización

[RF 13] – Crear nuevas organizaciones

El Admin podrá crear nuevas organizaciones.

*Entrada:*

El Admin pulsa sobre la pestaña “Home” > “Organizations” o directamente sobre la pestaña “Administrator” y rellena los campos “Name”, “Organization” y “Mail” y pulsa sobre “Create Organization”

*Sistema:*

El Sistema recoge los datos introducidos por el usuario y ejecuta una consulta a la base de datos para crear un nuevo registro con esa organización.

*Salida:*

Si todo fue bien, el Sistema añade esa nueva organización.

Si no, indica el error producido.

#### [RF 14] – Consultar información de los usuarios

El Admin podrá consultar la información asociada a cada una de los usuarios.

*Entrada:*

El Admin pulsa sobre la pestaña “Home” > “Organizations” o directamente sobre la pestaña “Administrator” para poder ver todas las organizaciones, pudiendo ser filtrados por: Organización y Rol.

*Sistema:*

El Sistema ejecuta una consulta a la base de datos para obtener una tabla de todos los usuarios existentes.

*Salida:*

El Sistema, por defecto, muestra una tabla con la información de cada usuario.

#### [RF 15] – Crear nuevos usuarios

El Admin podrá crear nuevas usuarios.

*Entrada:*

El usuario pulsa sobre la pestaña “Home” > “Organizations” > “Users” o directamente sobre la pestaña “Administrator” > “Users”, selecciona la “organización” y el “rol”, pulsa sobre “Add User” y rellena el formulario.

*Sistema:*

El Sistema recoge los datos introducidos y ejecuta una consulta a la base de datos para crear un nuevo registro con es usuario.

*Salida:*

Si todo fue bien, el Sistema añade ese nuevo usuario.

Si no, indica el error producido.

#### [RF 16] – Consultar Base de Datos

El Admin podrá consultar la base de datos de la aplicación.

##### *Entrada:*

El usuario pulsa sobre la pestaña “Database” y con el desplegable selecciona por qué campo quiere filtrar.

##### *Sistema:*

El Sistema recoge la opción elegida y ejecuta una consulta a la base de datos.

##### *Salida:*

El Sistema muestra la base de datos filtrada por ese campo.

## **Requisitos no funcionales**

Los requisitos no funcionales se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar, sino características de funcionamiento. A continuación se muestran en función a distintas partes de la aplicación.

### ***Documentación***

#### [RNF 1] - Documentation del Sistema completo

Se elaborará una documentación completa y detallada de todo el sistema a nivel de código, técnico (descripción y documentación del código, sus atributos y funciones y sus módulos o clases) y de las herramientas que se usen. Todo esto será útil si en algún momento del ciclo de vida de este software el cliente desea realizar alguna modificación o se decide vender sus derechos.

#### [RNF 2] - Ayuda al usuario

La aplicación dispondrá de un botón de ayuda, llamado “Feedback”, presente en cada una de las ventanas que permitirá al usuario realizar preguntas, reportar problemas, sugerir mejoras y decir qué le gusta de la aplicación y su contenido. Dichas acciones podrán realizarse adjuntados archivos de texto y multimedia.



## ***Rendimiento***

### **[RNF 3] - Rapidez de navegación**

La aplicación se usará en países en los que el acceso a Internet y la velocidad del mismo sean muy limitados, por lo que el tiempo de carga entre las distintas interfaces de la aplicación deberá ser el menor posible.

### **[RNF 4] - Soporte para todos los usuarios**

Debido a la gran cantidad de usuarios que podrá acceder a la aplicación, esta deberá poder mantener un rendimiento óptimo en cualquiera de los dispositivos desde los que se ejecute.

## ***Disponibilidad y accesibilidad***

### **[RNF 5] - Disponibilidad 24/7**

La aplicación será accesible vía web (alojada en los servidores OVH de la empresa) para cualquier dispositivo, de modo que se elija el formato que se elija, la aplicación tendría una disponibilidad de 24/7 siempre que los servidores estén activos, que excepto en breves periodos de mantenimiento, lo estarán.

En caso de lugares con conexión limitada, la aplicación será accesible pero con funcionalidad reducida.

## ***Seguridad***

### **[RNF 6] - Claves de acceso privadas**

Las claves de acceso a la plataforma de cada usuario serán privadas en todo momento, el administrador no tendrá forma de conocerlas.

### **[RNF 7] - Seguridad en el sistema**

El sistema tendrá un nivel de seguridad razonable y acorde a su funcionalidad para evitar ataques y especialmente una seguridad efectiva que proteja su base de datos de ataques por ejemplo de SQL Injections o ingresos fraudulentos en el LogIn.

## ***Usabilidad***

### **[RNF 8] - Recordar cuenta de usuario en terminales**

Se dará la opción al usuario de recordar su usuario y contraseña en todo terminal desde el que inicie sesión.

#### [RNF 9] - Heurísticas de Nielsen

Basándonos en las heurísticas de Nielsen, esperamos tener una usabilidad aceptable, es decir, de al menos el 70%. Esto implica tener en cuenta aspectos como que la aplicación sea sencilla e intuitiva para el usuario, informar en todo momento del estado de la misma durante su uso, que siga los estándares establecidos, prevención de errores, etc.

#### [RNF 10] - Idiomas

La aplicación estará disponible al menos en los idiomas Inglés, Francés y Español.

#### ***Estabilidad***

#### [RNF 11] - Aplicación consistente

Se espera que la aplicación sea estable y sin errores, para conseguirlo, durante su desarrollo y especialmente al final, se le realizarán una serie de baterías de pruebas suficientes como para asegurar un funcionamiento eficiente.

#### ***Portabilidad***

#### [RNF 12] - Aplicación multiplataforma

Accesible vía web desde cualquier navegador y con dispositivos con cualquier sistema operativo (Android, OSx, Windows Phone y BlackBerry Mobile)

#### ***Operatividad***

#### [RNF 13] - Operatividad adaptada

Su operatividad está pensada dentro del ámbito de la Organización contratante, es decir, podrá ser usada por cualquier persona sin la necesidad de tener conocimientos previos de informática.

## ***Mantenibilidad***

### **[RNF 14] - Mantenimiento y modificaciones**

El sistema será altamente mantenible puesto que el esfuerzo asociado a su funcionamiento normal o para restituirlo si se presentara una situación de fallo, debería ser relativamente bajo. Además será suficientemente flexible para permitir ciertas modificaciones si fueran necesarias en el futuro por parte de los técnicos o incluso por parte del administrador general.

## ***Interfaz***

### **[RNF 15] - Sencilla e intuitiva**

Será desktop first y garantizará al usuario una correcta, fácil e intuitiva visualización de la interfaz de la aplicación, independientemente del tamaño de la pantalla que use el dispositivo desde el cual se accede a la aplicación y de sus conocimientos informáticos.

## B Feedback con el cliente

- Google Docs

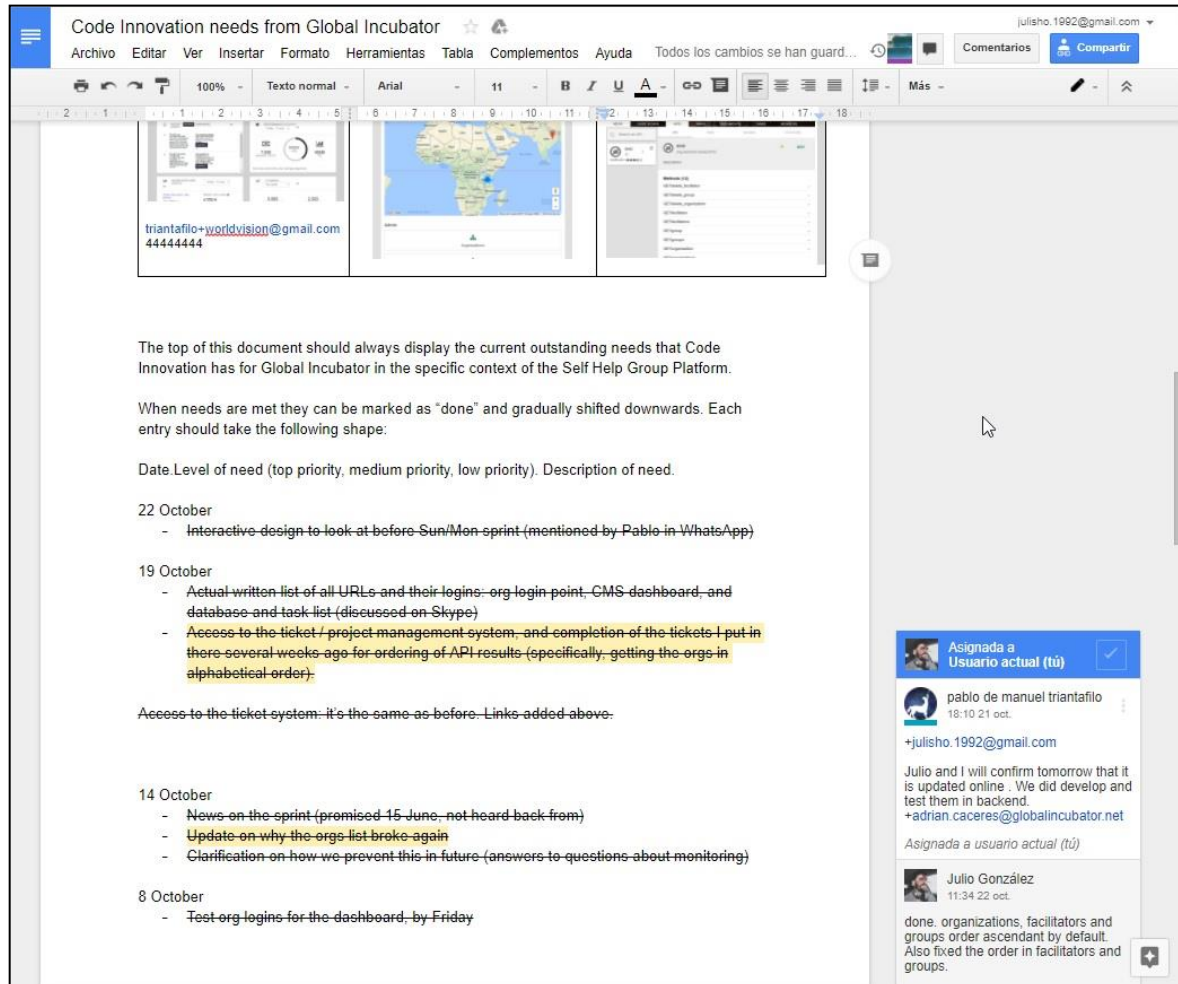


FIGURA 15: SPRINT CON LOS MIEMBROS DEL EQUIPO

En la imagen se muestran algunos de los feedbacks y patrones de diseño que el cliente me proporcionó durante el desarrollo del proyecto.

Screen	Content	Corrected
All screens	Text / prev	Can we change these to just icons, the "<" and ">" from the Material icon set. Thanks! <a href="https://material.io/icons/">https://material.io/icons/</a>
Home	<p>Your home is your main dashboard</p> <p>You can see your key aggregated numbers and even set a goal, for instance in savings and visualize your progress</p> <p>You will also see automatic alerts, as well as communications from code reservation and they have feedback, issues solved and updates with content or functionality</p>	<p>Your home is your main dashboard</p> <p>You can see your key aggregated numbers, <del>set goals</del>, and visualize your progress</p> <p>You will also see <del>automatic alerts</del> and updates from the development team, providing personalised feedback for your organization, as well as updates to the SHG app's content and functionality</p>
Manage Your Facilitators	<p>Manage Your Facilitators</p> <p>Supervise <del>their</del> progress, understand their averages.</p> <p>IN THE NEAR FUTURE</p> <ul style="list-style-type: none"> <li>- We have created a smart push notification system so that the may receive alerts directly to their mobile phone.</li> <li>- Invite a new one through email or phone number</li> </ul>	<p>Manage Your Facilitators</p> <p>Supervise <del>their</del> progress, understand their averages.</p> <p>In the near future: (Please use this in bold or a heading, not all caps)</p> <ul style="list-style-type: none"> <li>- We have created a smart push notification system so that facilitators can receive alerts directly on their mobile phone.</li> <li>- Invite a new facilitator through email or phone number</li> </ul>

FIGURA 16: CORRECCIONES DEL CLIENTE SOBRE EL DISEÑO

SHG project style and UX guide

Archivo Editar Ver Herramientas Ayuda

100%

Solo lectura

Compartir

Colors

High-contrast colours are key in this project as the app and associated resources are often used in less than ideal lighting. Many facilitators use the app during meetings outdoors.

We have a set palette of colors for the app. They currently are:

White	#ffffff		This should be the background of all large sections of text, and the font color when text is over dark colours.
Black	#000000		The font colour for all large text sections.
Darkest grey-blue	#09091a		Menu / top bar color (with white text)
Dark grey-blue	#222233		Button color either on white on on the darkest grey-blue
Red	#c41204		Button / important call to action / cancel colour
Dark Grey	#424242		Sub-heading colour (eg, the unit numbers in the app curriculum are in this colour)
Light grey	#d2d2d2		Background colour in emphasis areas

FIGURA 17: PATRONES DE DISEÑO DEL CLIENTE

- WhatsApp

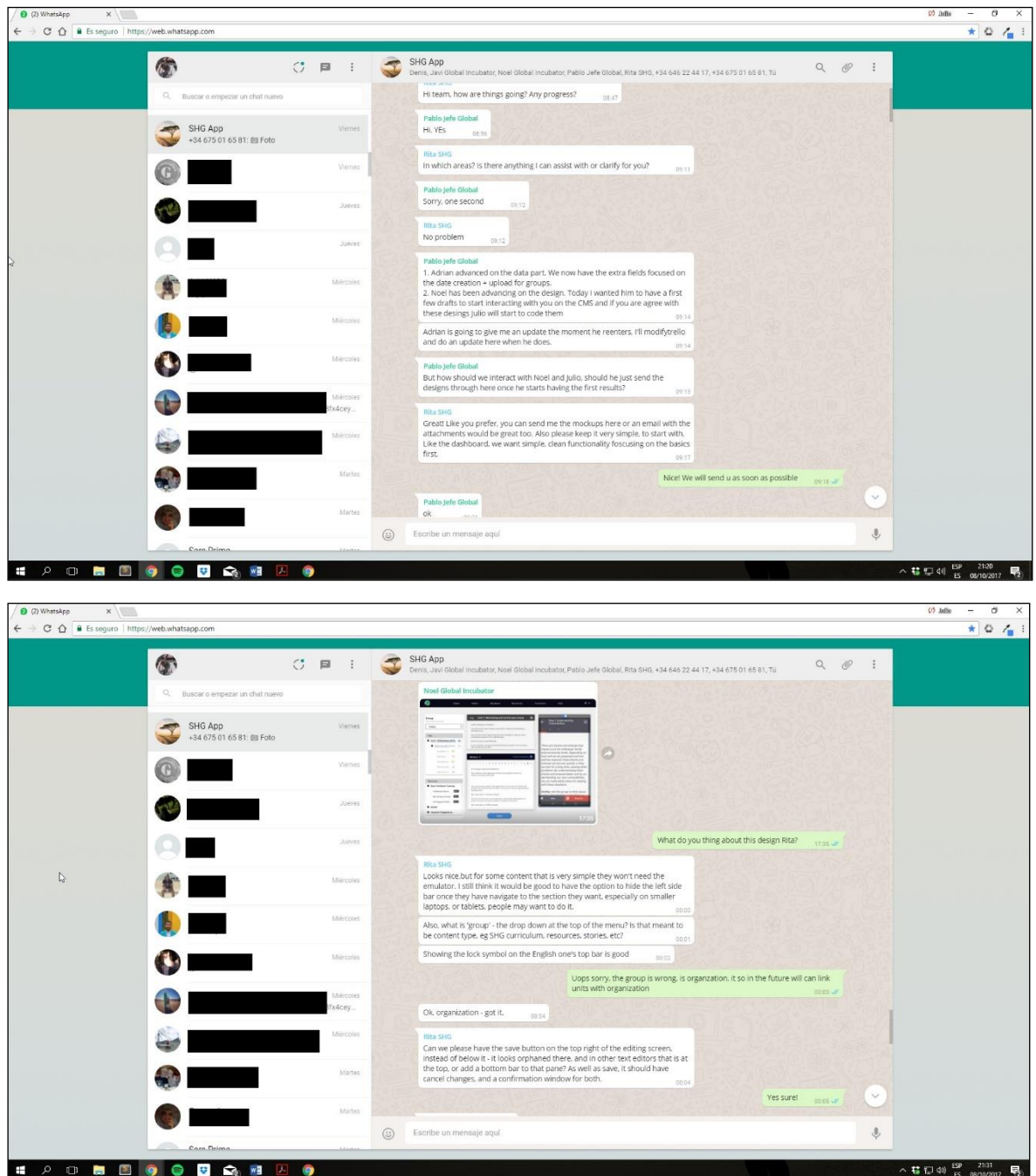


FIGURA 18: FEEDBACK CON EL CLIENTE A TRAVÉS DE WHATSAPP

- Trello

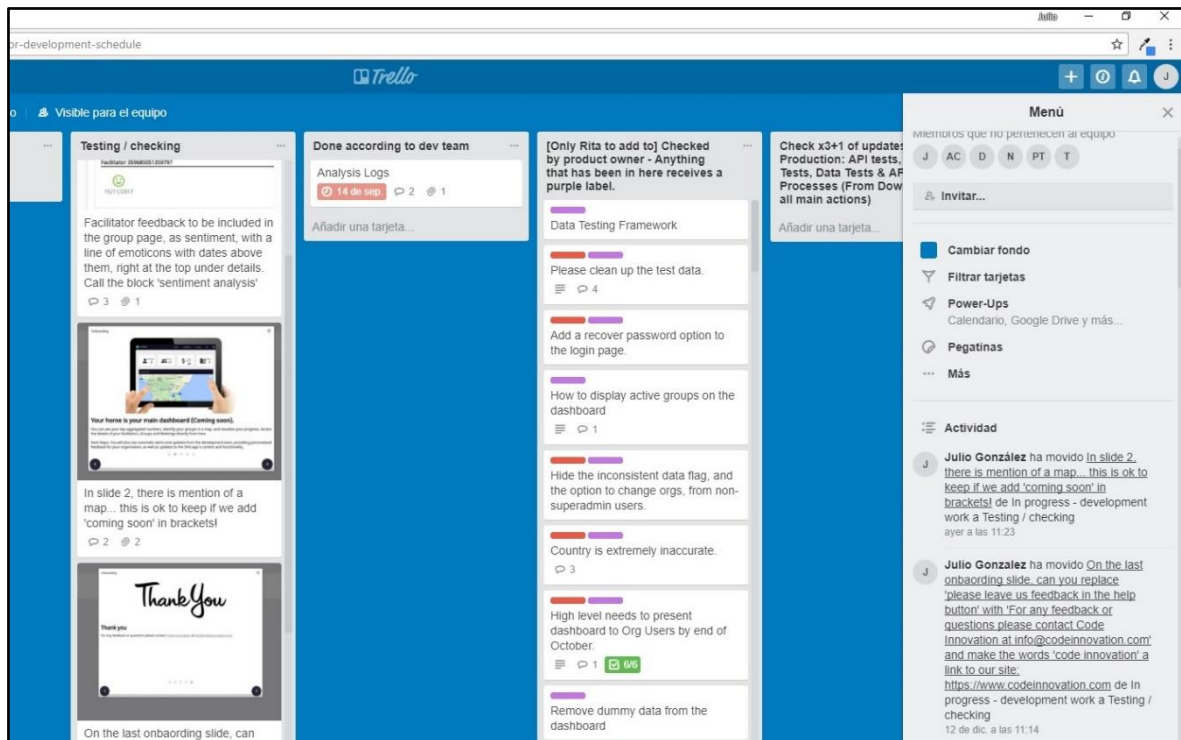


FIGURA 19: FEEDBACK CON EL CLIENTE A TRAVÉS DE TRELLO

## C Pruebas realizadas

### 1. Log-In

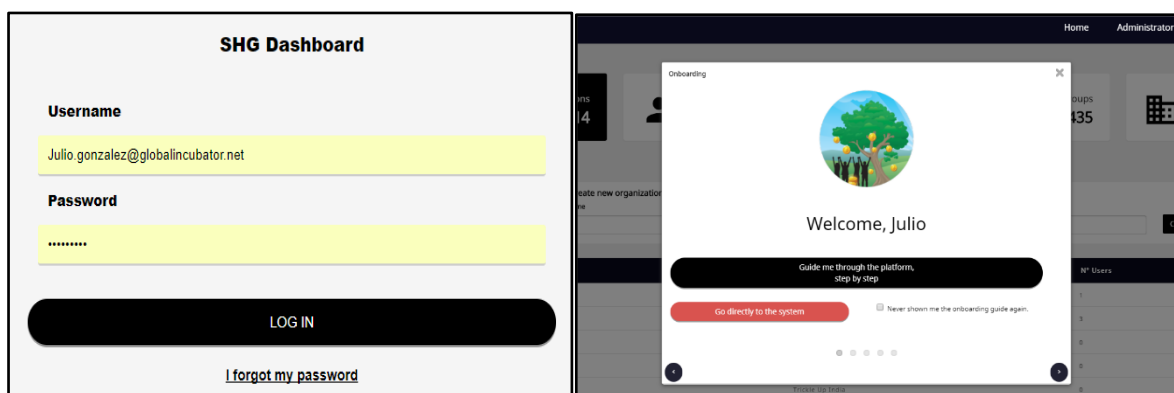


FIGURA 20: PRUEBA LOG IN

## 2. Cambio de idioma (Inglés a Francés)

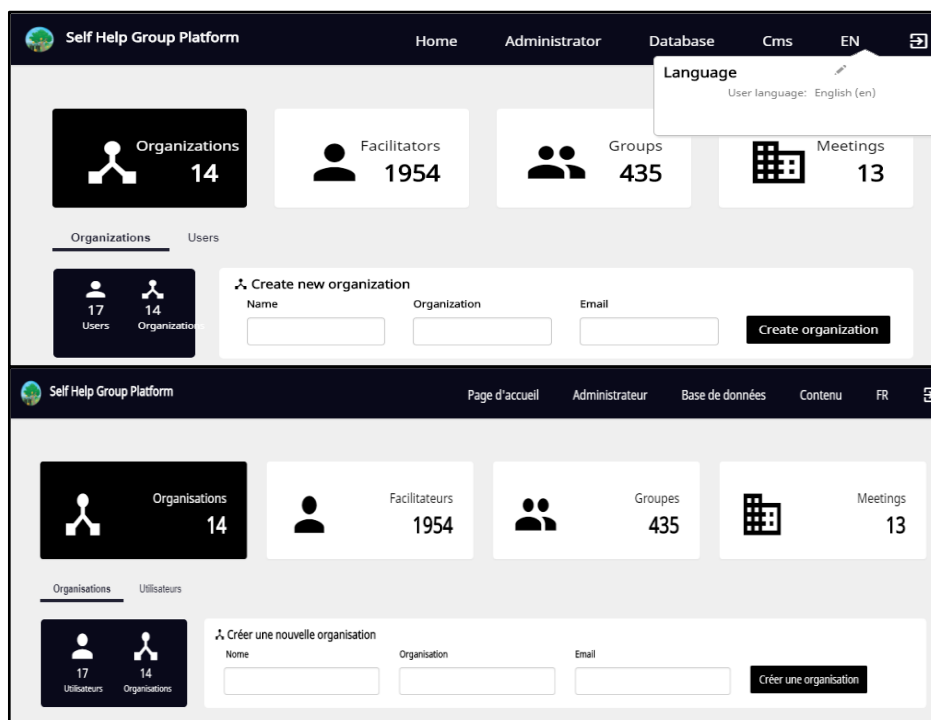


FIGURA 21: CAMBIO DE IDIOMA

## 3. Editar coordenadas de un grupo

Title	Group ID
PAMOJA ONE HEN	49
PAMOJA ONE HEN	60
yataka moyo	333
PAMOJA ONE HEN	70
PAMOJA ONE HEN	74
PAMOJA ONE HEN	47
PAMOJA ONE HEN	58
PAMOJA ONE HEN	65
PAMOJA ONE HEN	59
PAMOJA ONE HEN	51
TUUNGANE PAMOJA	73
TUUNGANE PAMOJA	72
MSAMALIA MWEMA	71
FARAJA	66
AMANI PAMOJA	77

### AMANI PAMOJA

Map showing the location of Amani Pamoja near Dar es Salaam, Tanzania. The map includes labels for various locations like Kilosa, Morogoro, and Dar es Salaam, as well as geographical features like Mikumi National Park and the Indian Ocean.

**Edit location**

Click on the map with the right mouse button to select some coordinates or enter them manually

Lat:  Lng:

**Save**

Last Access	View
3 years ago	<a href="#">View</a>
3 years ago	<a href="#">View</a>
6 months ago	<a href="#">View</a>
2 years ago	<a href="#">View</a>
2 years ago	<a href="#">View</a>
2 years ago	<a href="#">View</a>
3 years ago	<a href="#">View</a>
3 years ago	<a href="#">View</a>
3 years ago	<a href="#">View</a>
3 years ago	<a href="#">View</a>
2 years ago	<a href="#">View</a>
3 years ago	<a href="#">View</a>
3 years ago	<a href="#">View</a>
3 years ago	<a href="#">View</a>
3 years ago	<a href="#">View</a>
3 years ago	<a href="#">View</a>



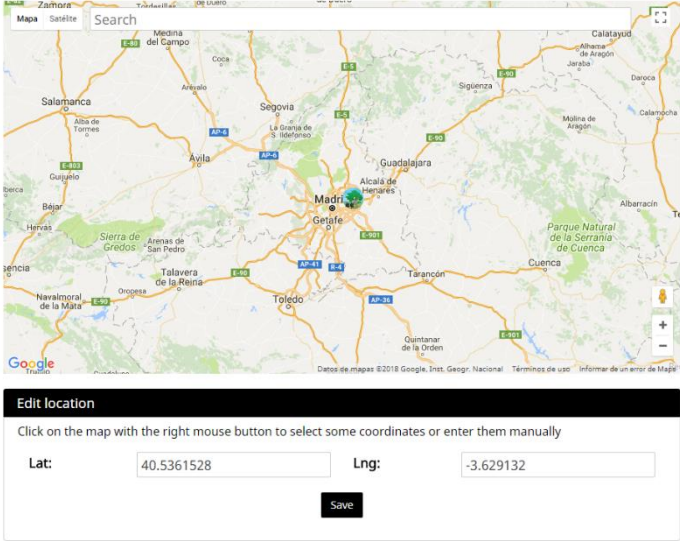
Title	Group ID	AMANI PAMOJA	Last Access	View
PAMOJA ONE HEN	70		2 years ago	<a href="#">View</a>
PAMOJA ONE HEN	47		2 years ago	<a href="#">View</a>
PAMOJA ONE HEN	74		2 years ago	<a href="#">View</a>
PAMOJA ONE HEN	51		2 years ago	<a href="#">View</a>
AMKA	68		2 years ago	<a href="#">View</a>
AMANI PAMOJA	77		3 years ago	<a href="#">View</a>
FARAJA	66		3 years ago	<a href="#">View</a>
TUUNGANE PAMOJA	73		3 years ago	<a href="#">View</a>
TUUNGANE PAMOJA	72		3 years ago	<a href="#">View</a>
MSAMALIA MWEMA	71		3 years ago	<a href="#">View</a>
PAMOJA ONE HEN	65		3 years ago	<a href="#">View</a>
PAMOJA ONE HEN	62		3 years ago	<a href="#">View</a>
PAMOJA ONE HEN	60		3 years ago	<a href="#">View</a>
PAAMOJA ONE HEN	59		3 years ago	<a href="#">View</a>
PAMOJA ONE HEN	58		3 years ago	<a href="#">View</a>

FIGURA 22: CAMBIO DE COORDENADAS

#### 4. Crear nuevo facilitador

The image consists of two screenshots of the 'Self Help Group Platform' interface, demonstrating the process of creating a new user.

**Top Screenshot:** The interface shows the 'Users' tab selected. The 'Create new user' form is visible, with 'Organization' set to 'Global Incubator' and 'Role' set to 'Facilitator'. A modal titled 'Crear Nuevos Usuarios' (Create New Users) is open, showing a list of users to be created. The modal includes a search bar, a list of users, and a 'Create users' button. A red arrow points to the '17 Users' button in the top left corner.

**Bottom Screenshot:** The interface shows the 'Users' tab selected. The 'Create new user' form is visible, with 'Organization' set to '-Select organization-' and 'Role' set to '-Select role-'. A red arrow points to the '18 Users' button in the top left corner. Below the form, a table lists the created users:

User	Role
Test Tester_1	SHG - Facilitator
Plan International Admin *	SHG - Organization
Nathaniel *	SHG - Admin
Denis vaillo *	SHG - Facilitator

FIGURA 23: CREAR NUEVO USUARIO

## ***D Sintaxis básica de AML***

Las sentencias de código AML están formadas por *palabras clave* y *atributos*, que varían en función del tipo de AML que usemos. A continuación, mostraré un ejemplo en la siguiente figura, donde la sentencia AML se encarga de mostrar un “panorama”, que viene a ser una tabla de información.

- **Palabras clave**

La palabra clave del ejemplo mostrado es la de color naranja, que viene a ser el atributo principal de la sentencia AML. Las sentencias AML pueden tener o no tener palabra clave, depende de su funcionalidad. El resto de atributos que usa la sentencia vienen precedidos del símbolo numeral #.

Pero hay sentencias en AML que no necesitan de palabras claves o atributos, por ejemplo es el caso de *client\_type*, que realiza una consulta sobre la base de datos y devuelve qué tipo de cliente tiene el panorama sobre el cual se realizó dicha sentencia AML.

- **Atributos**

Los atributos mostrados en el ejemplo corresponderían a las siguientes palabras: *type*, *position*, *resource*, *f*, *panoid*, *show\_filter* y *filter\_operations*, todas ellas precedidas del símbolo # que sirve para indicar que se tratan de atributos de esa sentencia AML. Dichos atributos son los parámetros de entrada que la sentencia AML tendrá que tener en cuenta al ejecutarse.

Al igual que con las palabras clave pueden haber sentencias AML sin atributos, pero lo más habitual es que las sentencias tengan una palabras clave y varios atributos.

- **Bucles**

Este tipo de acciones requiere especial atención dentro del pseudo-lenguaje AML, ya que para realizar bucles se utiliza la palabra clave *iterator*, que irá acompañada de una serie de atributos en función del comportamiento que queramos obtener.

Se pueden anidar tantos bucles como se quiera, pero hay que tener en cuenta que el código AML que vaya dentro de dichos bucles debe de ponerse de forma *parseada* y no como un AML original. Las equivalencias se muestran en la tabla.

AML original	AML parseado
[[	{<}
]]	{>}
{	{<-}
}	{->}
#	{*}
:	{:}

## ***E Ejemplos de tipos de AML***

De la gran cantidad de tipos de AML que hay, voy a comentar una serie de las funcionalidades más características y usadas en este pseudo-lenguaje

- “**context**”, que puede tomar varias funcionalidades

- reload: recarga un contexto

```
[[context:reload#f:formid#ctxt:ctxtID#fhtmls:fhtmlsIDS#core_questions:core_questions#target:target]]
```

Example:

```
[[context:reload#f:944#ctxt:225588#fhtmls:11,22#core_questions:1#target:fHTML.fHTML-11,questions.3333.value]]
```

Attributes:

| core\_questions\* - 1 o 0 para indicar si recalculamos las respuestas de ese contexto, por defecto es 1

| fhtmls\* - ids de los fhtmls que queremos recargar separados por comas, si no se le pasa este parametro se recargan todos los fhtmls

| target\* - se puede indicar que devuelva los atributos que queramos, se indican separados por comas

*Return:* si se especifica el atributo target devuelve lo que se le indique

- delete: Borra un contexto

*Template:* [[context:delete#f:formid#ctxt:ctxtID]]

*Example:* [[context:delete#f:999#ctxt:44444]]

*Attributes:*

| ctxt - id del contexto a borrar

| f - id del formulario

*Return:*

| invalid\_context\_id – si no se pasa un contexto

| fmo\_permisos\_qapp - si no se tiene permisos de edición

| error\_deleting\_context - si el contexto está ya borrado

- fold: Obtiene datos de un formulario filtrando y ordenando por alguna de sus columnas, se pueden hacer diferentes operaciones con los datos.

*Template:*

[[context:fold#f:formid#panoid:panoramaID#op:operation#target:target#filter\_operations:filters#order\_operation:order#load\_all\_columns:0#load\_free\_columns:0#size:size#offset:offset]]

*Example:*

[[context:fold#f:944#panoid:12313#op:concat#target:questions.2222.value#filter\_operations:{"type":"filter","target":"initiative","value":222,"ctxt":"in"}-{"type":"filter","target":"contact","value":[333,555],"ctxt":"in"}#order\_operation:{"target":"questions.2222.value","ctxt":"ASC"}#load\_all\_columns:0#load\_free\_columns:0#size:5#offset:0]]

*Attributes:*

| panoid - id del panorama

| target - atributo sobre el que se realiza las operaciones, en el caso de la operación count o es necesario, en el caso de concat y json se pueden poner varios atributos separados por comas.

| f - id del formulario

| load\_free\_columns\* - 1 o 0 si para la operación no se necesita ningún fHTML tipo 3 o 2.5 es recomendable ponerlo a 0 para optimizar la velocidad del fold, por defecto es 1

| order\_operation\* - orden que queramos aplicar, {"target":"questions.2222.value","ctxt":"ASC"}

| filter\_operations\* - filtros que queramos aplicar con el siguiente formato: {"type":"filter","target":"initiative","value":222,"ctxt":"in"}

| offset\* - número de página que queramos obtener, depende del parámetro size para saber a partir de que elemento traer, por defecto es 0

| load\_all\_columns\* - 1 o 0 si se define las columnas que se descargan un panorama se puede poner a 0 para optimizar la velocidad de fold, por defecto es 1

| size\* - número de elementos a devolver, si no se define devuelve todos

| op - operación que se realiza con los datos, posibles valores: count,count\_distinct,concat,concat\_distinct,json,avg,sum,max,min,product,stddev,var

*Return:*

| invalid\_context\_id – si no se pasa un contexto

| fmo\_permisos\_qapp - si no se tiene permisos de edición

| error\_deleting\_context - si el contexto está ya borrado

- delete: Borra un contexto

*Template:* [[context:delete#f:formid#ctxt:ctxtID]]

*Example:* [[context:delete#f:999#ctxt:44444]]

*Attributes:*

| ctxt - id del contexto a borrar

| f - id del formulario

*Return:*

| invalid\_context\_id – si no se pasa un contexto

| fmo\_permisos\_qapp - si no se tiene permisos de edición

| error\_deleting\_context - si el contexto está ya borrado

- “**utils**”, que puede tomar varias funcionalidades
  - optional\_param: Devuelve el valor de una variable pasada por GET o POST  
*Template:* [[utils:optional\_param#p:var\_name#default\_value:default\_value]]  
*Example:* [[utils:optional\_param#p:id\_user#default\_value:123]]  
*Attributes:*
    - | p - Nombre de la variable
    - | default\_value\* - Valor de la variable en caso de no ser recibida
  - get\_urls: Devuelve toda las url que encuentra en un texto  
*Template:* [[utils:get\_urls#str:text#only\_domains:1]]  
*Example:* [[utils:get\_urls#str:var\_text#only\_domains:0]]  
*Attributes:*
    - | str - Texto en el que va a buscar
    - | only\_domains\* - Indica si va a devolver solo los dominios o las urls completas
  - is\_mobile: Devuelve si el usuario está navegando con un dispositivo movil  
*Template:* [[utils:is\_mobile]]  
*Example:* [[utils:is\_mobile]]  
*Attributes:* no necesita
  - sleep: Mantiene el script en espera durante X segundos  
*Template:* [[utils:sleep#time:seconds]]  
*Example:* [[utils:sleep#time:10]]  
*Attributes:*
    - | seconds - Segundos que se va a mantener en espera el script

## F Ejemplo de uso de QApp

A continuación se muestra en la <figuraX1> todas las QApp pertenecientes a la aplicación y en la <figuraX2> un ejemplo de cómo es la interfaz para la creación de un QApp:

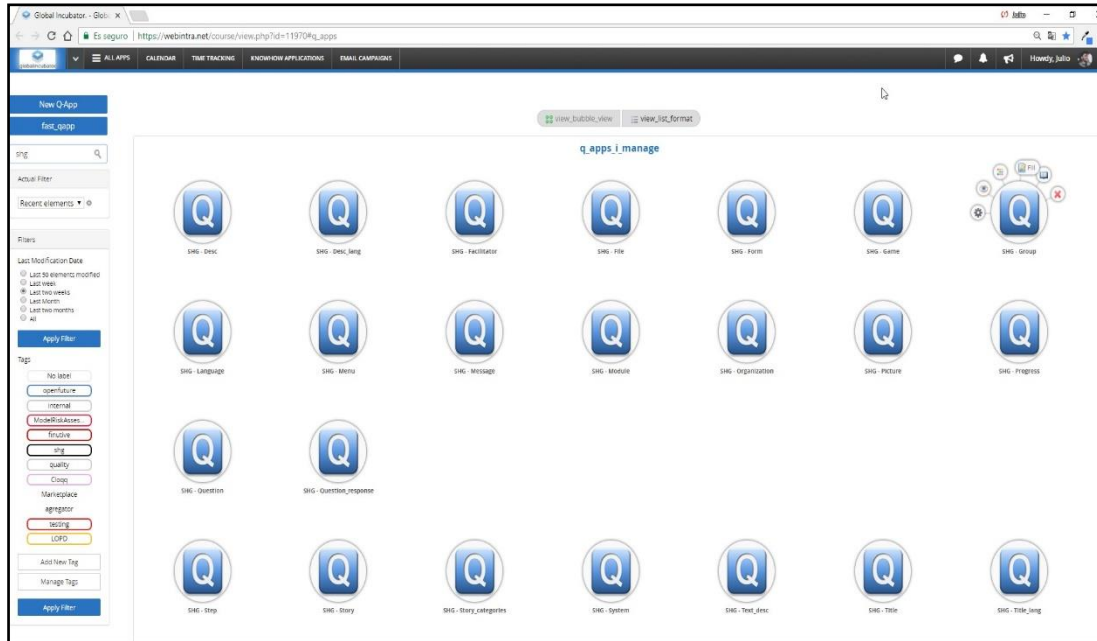


FIGURA 24: MENU 1 DE QAPPS



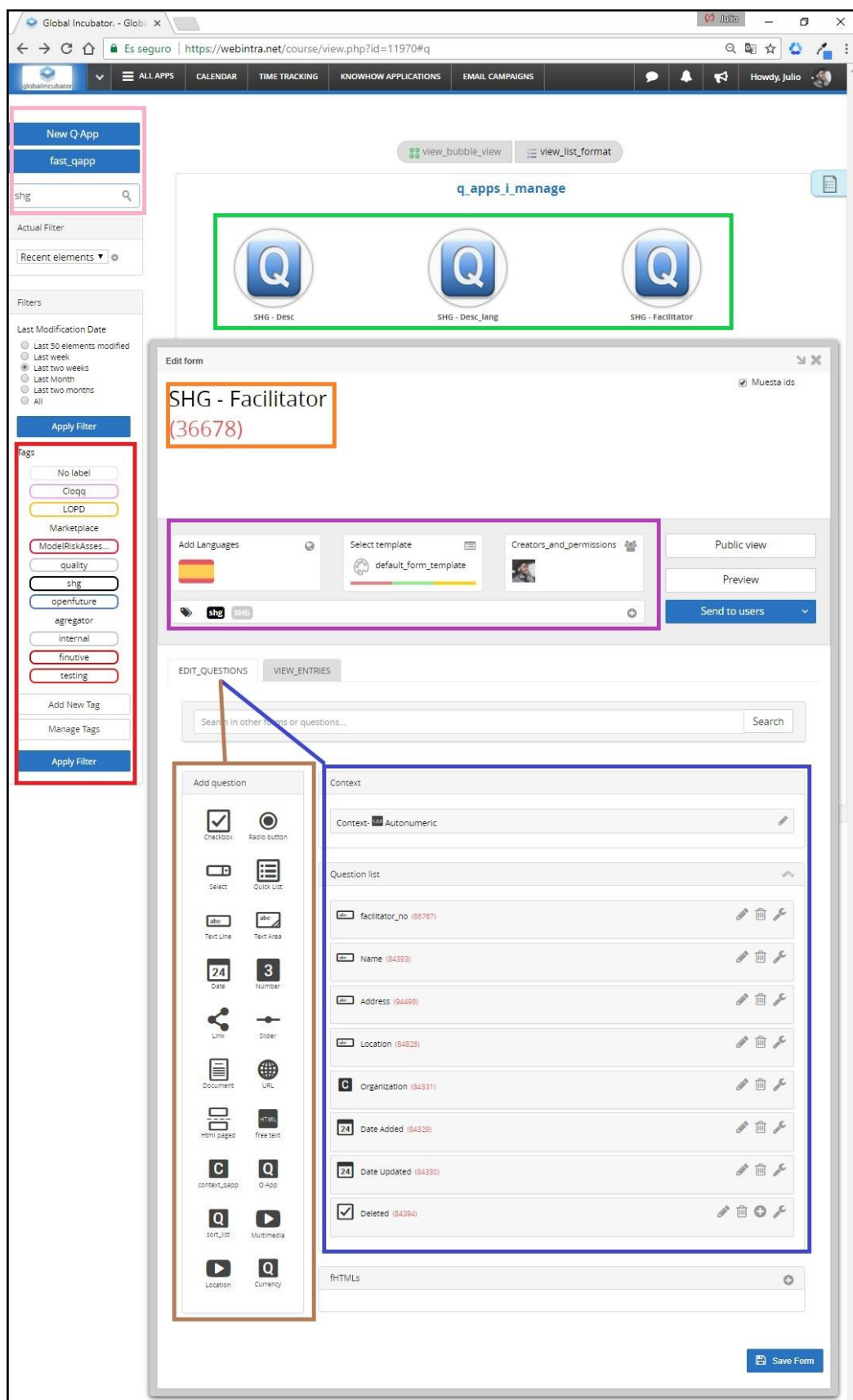


FIGURA 25: MENÚ 2 DE QAPPS

En el recuadro rosa podemos ver los botones *New Q-App* y *fast-qapp* y el buscador de QApp, que nos permite buscar cualquier QApp de manera más rápida. También podemos buscar QApps filtrando por su fecha de creación y sus Tags asociados, como se muestra en el recuadro rojo. En el recuadro verde podemos ver los QApps que ya tenemos creado previamente o que ha creado otro usuario pero tenemos derecho para su edición.

Si pulsamos alguno de los dos botones del recuadro rosa se nos abrirá el popup *Edit form* que cuenta con un campo para poner el título del QApp que estamos creando y un ID (recuadro naranja) que se asigna de forma automática por el sistema y que será la referencia a dicho QApp en caso de que queramos acceder a él (por ejemplo realizando un *[[context.fold...]]* con una sentencia AML ). El recuadro morado nos da la opción de elegir el idioma del formulario, el template que queremos usar para su visualización y la información sobre los creadores y sus permisos.

Más abajo, tenemos las pestañas *EDIT\_QUESTIONS* y *VIEW\_ENTRIES*, si pulsamos sobre la primera se nos mostrarán los recuadros marrón y azul. En el recuadro marrón tenemos una amplia variedad de preguntas para añadir a nuestro formulario (textarea, radio, links a YouTube, select, etc), por otro lado, en el recuadro azul tenemos el apartado *Context* y *Question List*.

Hay que decir que un QApp puede tener como campo una pregunta de tipo QApp, permitiendo de este modo relacionar QApps entre sí.

Los QApps están caracterizados por su contexto, que determinará la forma en la que los formularios serán generados y guardados en la base de datos. Hay varios tipos de contexto, los más usados son:

- Autonumeric: el id del formulario se irá incrementando a medida que reciba entradas.
- Initiative: está pensado para uso de “empresa”, ya que permite que sólo un usuario de cada empresa tenga acceso para rellenarlo.
- User: cada usuario del sistema podrá acceder a un solo formulario que registrará sus respuestas la próxima vez que entre.

- Role: se podrá acceder al formulario en función del rol que se tenga.

En el apartado *Question list*, se nos mostrarán las preguntas que ya hemos creado y vamos a añadir tras elegir las del recuadro marrón, donde podremos editarlas.

Si en la pestaña EDIT\_QUESTIONS podíamos elegir entre una amplia variedad de campos para añadir a nuestro formulario, ahora si pulsamos sobre la pestaña VIEW\_ENTRIES veremos las columnas de la base de datos que nosotros decidamos mostrar.

Para decidir sobre qué columnas queremos mostrar, en el recuadro rojo, pulsaremos sobre el icono del lapicero de “Ver con mis permisos”, el cual nos mostrará la ventana de la

Esta ventana tiene un contenedor (recuadro azul) con toda la posibilidad de columnas que podemos mostrar y de filtros que podemos usar.

SHG - Facilitator (36678)

Add Languages: Spanish

Select template: default\_form\_template

Creators\_and\_permissions: SHG

Public view: Preview

Send to users: Send to users

EDIT\_QUESTIONS VIEW\_ENTRIES

2009 Replies 5 / 5 Panoramas Create\_new\_panorama

Search here

HIDE FILTERS

HIDE PANORAMAS

Ver con mis permisos (96372) ✎ ★

Tabla de panoram... (96373) ✎ ★

Burbuja de panor... (96374) ✎ ★

Admin Database (96375) ✎ ★

Regular user dat... (119620) ✎ ★

Ver con mis permisos (0 Entries)

ID	INCONSISTENT DATA	FACILITATOR_NO	NAME	ORGANIZATION	DATE AD
(23268376) 23268376	No	58bd88880530276		World Vision Tanzania	2017-11-04 21:48:26
(23268375) 23268375	No	d605f10d9a32ac4		World Vision Kenya	2017-11-04 21:47:10
(23252219) 23252219	No	99459041a94c251		World Vision Kenya	2017-11-04 23:23:29
(23253449) 23253449	No	9b68bcbcd01d00f		World Vision Kenya	2017-11-04 04:48:28
(23273355) 23273355	No	f264f2ee5c35411		World Vision Kenya	2017-11-04 05:01:15
(17504189) 17504189	No	fd3048d12c6302c		World Vision	2017-07-15 09:41:41
(17543384) 17543384	No	303b23a7b28dbd4		World Vision	2017-07-21 14:34:03

FIGURA 26: MENÚ DE FACILITATOR

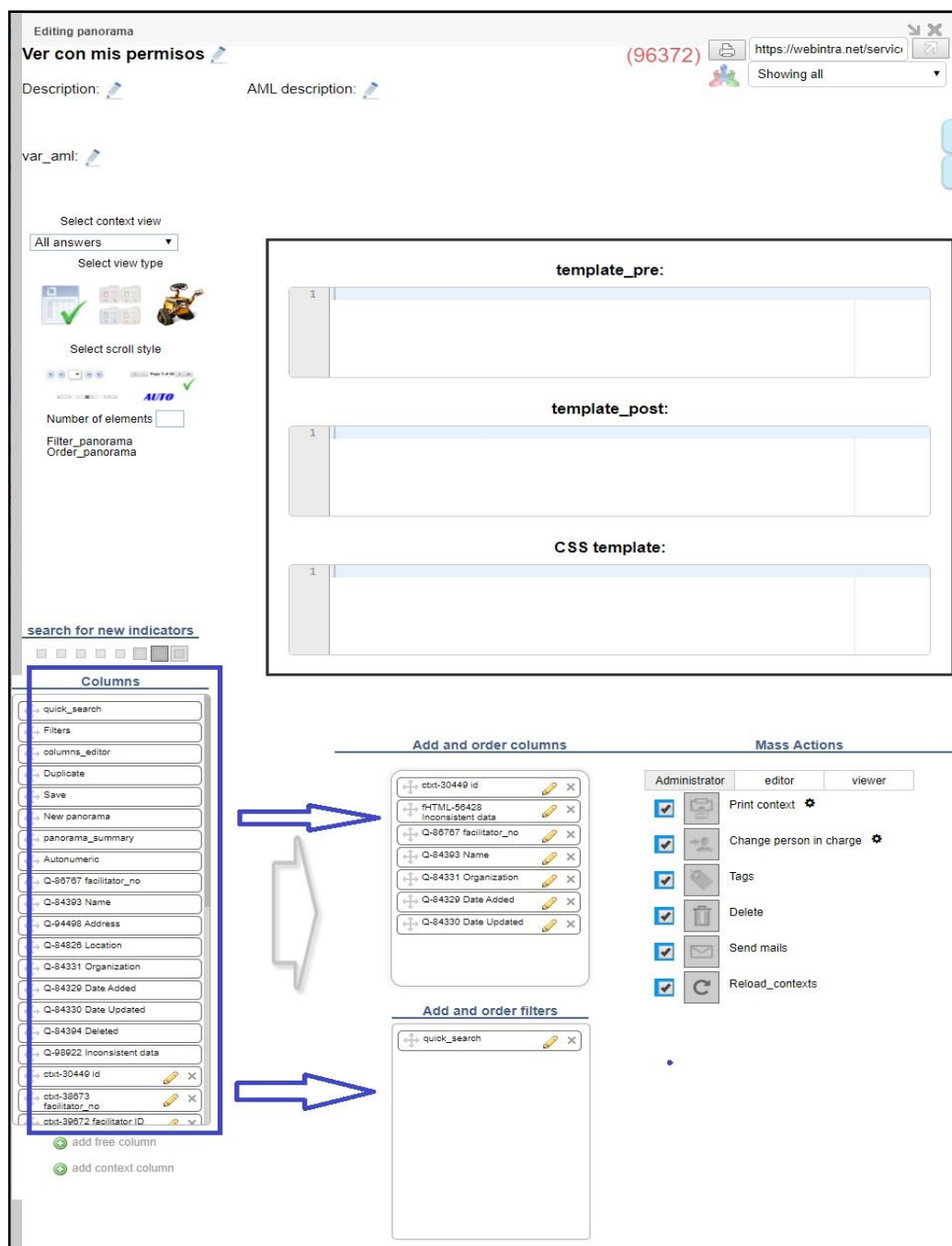


FIGURA 27: AÑADIR CUESTIONES

## G Ejemplo de uso de FreeApp

A continuación se muestra en la <figuraX1> un ejemplo de cómo es la interfaz de la App que proporciona el entorno de desarrollo de la aplicación.

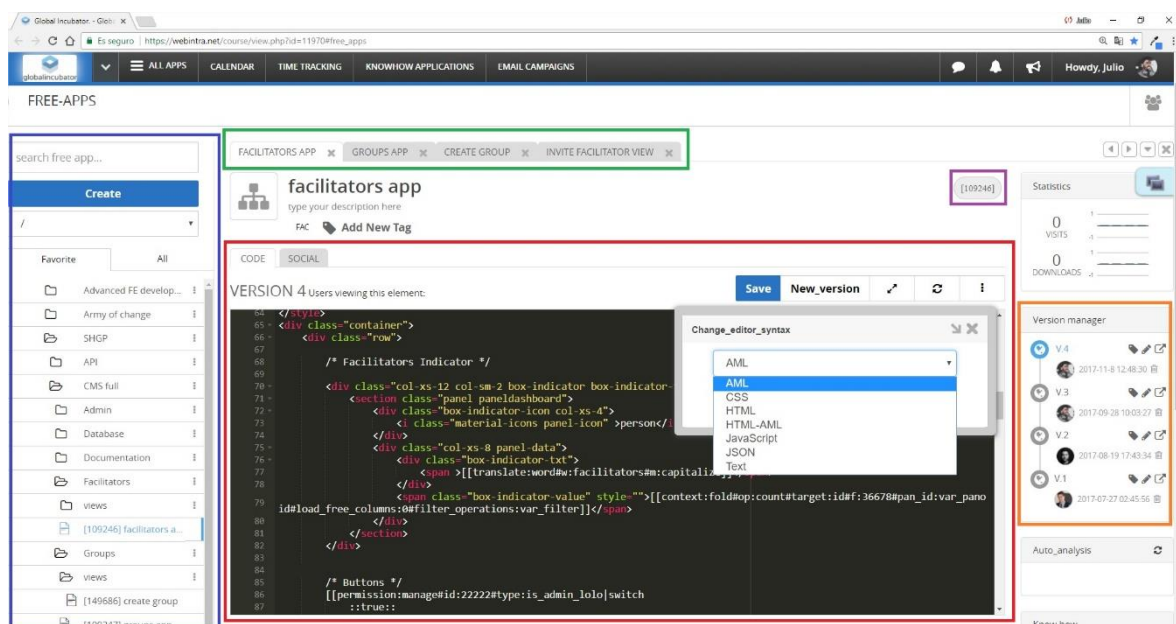


FIGURA 28: EJEMPLO DE USO DE FREEAPP

En el recuadro azul podemos ver la jerarquía de ficheros (freeApps) y carpetas con los que estamos trabajando y por los cuales podemos navegar. En este caso estamos dentro de la carpeta del proyecto SHG, donde podemos acceder a las distintas subcarpetas para crear, ver o editar nuevas freeApps. El código de identificación de cada freeApp aparece en el recuadro morado.

El recuadro verde nos muestra las pestañas con todas las freeApps que tenemos abiertas, sin importar del proyecto que sean. Y en función de la que seleccionemos, en el recuadro rojo se nos mostrará el código. A la hora de codificar, tendremos (de izquierda a derecha) las opciones de guardar el progreso mediante el botón “Save”, crear una nueva versión mediante el botón “New\_version”, expandir la ventana, actualizar el contenido y una serie de opciones como cambiar el tema del código, sintaxis, fuente, abrir el parser, etc. Dentro del tipo de sintaxis de código que reconoce la herramienta podemos encontrar los siguientes lenguajes: AML, CSS, HTML, HTML – AML, JavaScript, JSON y Text. En función de la sintaxis elegida el texto se mostrará en un color u otro, haciendo que la codificación del mismo resulte más fácil e intuitiva.

Finalmente, otra parte importante es el recuadro naranja ya que nos muestra el control de versiones con sus respectivas fechas de creación y autores. También nos permite, mediante el icono de la bola del mundo, saber si esa es la versión que está o no activa en la aplicación.

## H Parseador y ejemplos de AML

Global Incubator también desarrolló un parseador exclusivo de código AML para así poder testear de forma rápida cualquier sentencia AML. También soporta código HTML. A continuación se muestran una serie de imágenes con algunos ejemplos:

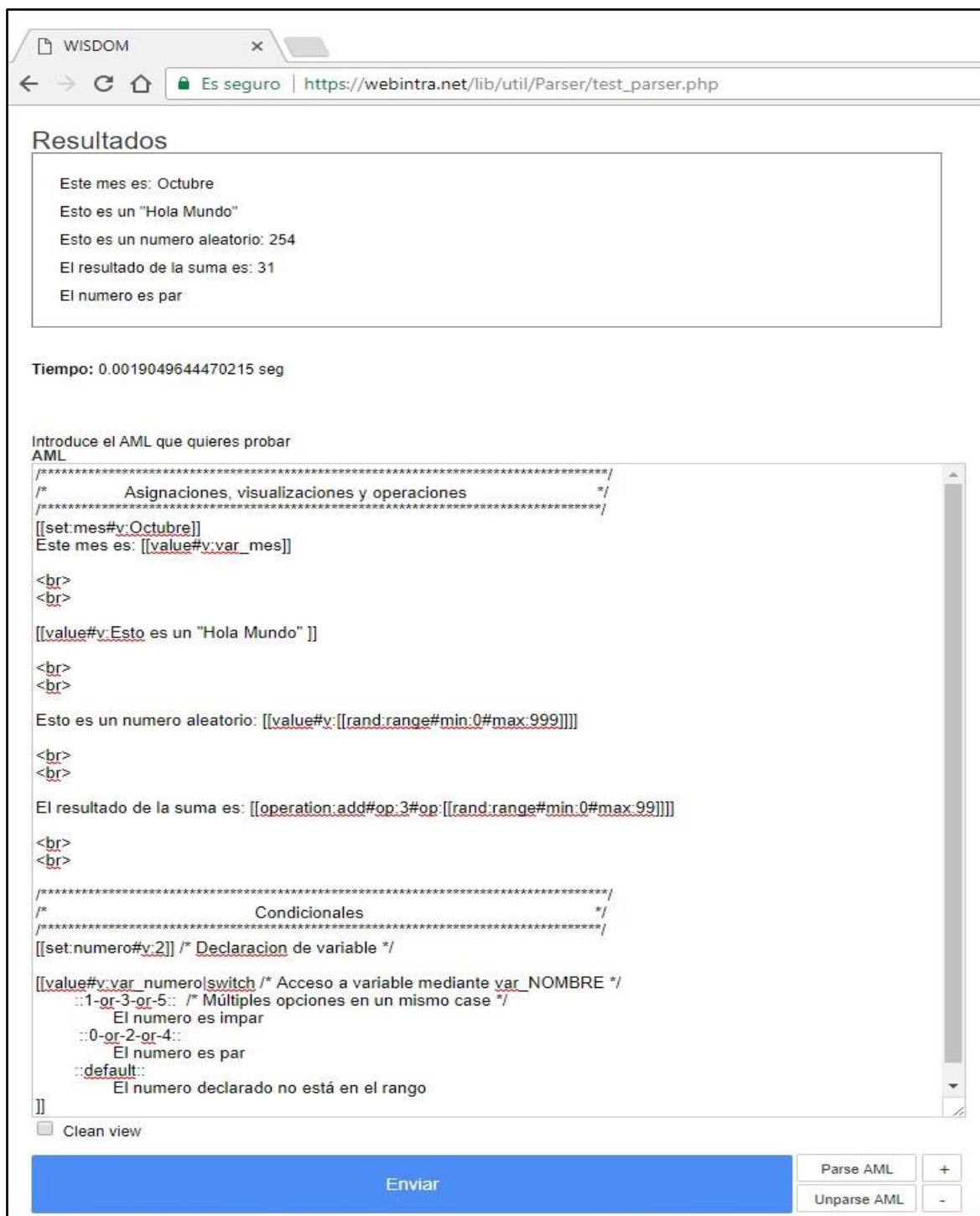


FIGURA 29: PARSEADOR AML



WISDOM
x

← → ↻ 🏠
Es seguro
https://webintra.net/lib/util/Parser/test\_parser.php
☆ 🔄 🛠

## Resultados

- Declaramos el array: [1,2,3,4]
- Insertamos un nuevo valor en el array: [1,2,3,4,"cinco"]
- Devolvemos el array como string: 1,2,3,4,cinco
- Concatenamos un valor a cada elemento: 1A2A3A4Acinco
- Cogemos el valor de la posición 2: 3
- Contamos el número de elementos del array: 5 elementos
- Buscamos el la posición de un elemento: El elemento 4 esta en la pos 3

- Definimos un objeto persona: {"name":"Juan","lastname":"Perez","edad":"22","casado":true}
- Mostramos el valor de una propiedad del objeto. La edad es: 22
- Borramos el valor de una propiedad del objeto. {"name":"Juan","edad":"22","casado":true}
- Operamos sobre valores del objeto: El doble de su edad es: 44 años.

Tiempo: 0.00097489356994629 seg

Introduce el AML que quieres probar

### AML

```

/*****
/*      Definición y acciones con arrays      */
*****/
[[array:numbers#a:set#v: [1,2,3,4] ]]
- Declaramos el array: [[array:numbers]]
<br>
- Insertamos un nuevo valor en el array: [[array:numbers#a:push#v:cinco ]]
[[array:numbers]]
<br>
- Devolvemos el array como string: [[array:numbers#a:join]]
<br>
- Concatenamos un valor a cada elemento: [[array:numbers#a:join#glue:A]]
<br>
- Cogemos el valor de la posición 2: [[array:numbers#a:get#p:2]]
<br>
- Contamos el número de elementos del array: [[array:numbers#a:count]] elementos
<br>
- Buscamos el la posición de un elemento: El elemento 4 esta en la pos [[array:numbers#a:search#v:4]]

/*****
/*      Definición y acciones con objetos      */
*****/
- Definimos un objeto persona:
[[class:persona name#v:Juan]]
[[class:persona lastname#v:Perez]]
[[class:persona edad#v:22]]
[[class:persona casado#o:true]]
[[class:persona]]
<br>
- Mostramos el valor de una propiedad del objeto. La edad es: [[class:persona.edad]]
- Borramos el valor de una propiedad del objeto. [[class:persona.lastname#a:unset]] [[class:persona]]
- Operamos sobre valores del objeto: El doble de su edad es: [[class:persona.edad#*.2]] [[class:persona.edad]] años.

```

☐ Clean view

Enviar

Parse AML +
Unparse AML -

FIGURA 30: EJEMPLO 2 DE PARSEADOR AML

WISDOM x

Es seguro | https://webintra.net/lib/util/Parser/test\_parser.php

## Resultados

- Declaracion e iteracion sobre un array simple  
Iteracion 0 // Valor: 1  
Iteracion 1 // Valor: dos  
Iteracion 2 // Valor: tres  
Iteracion 3 // Valor: 4
- Declaracion e iteracion sobre objetos:  
[{"fecha":1842,"titulo":"Amar","autor":"Juan Diez"}, {"fecha":1922,"titulo":"Odiar","autor":"Diego Diaz"}]  
El autor del libro 0 es Juan Diez  
El autor del libro 1 es Diego Diaz
- Iteracion con bucle tipo for  
Hola Mundo  
Hola Mundo  
Hola Mundo - iteracion con bucle tipo for + html  

Hola Mundo

Hola Mundo

Hola Mundo

### AML

```

/*
    Definición y tipos de iteradores
*/
- Declaracion e iteracion sobre un array simple
<br>
[[set numbers#v:1,dos,tres,4]]
[[iterator:simple#array:var_numbers#tmpl:
    Iteracion $0 //
    Valor: $1
    <br>
]]
[[array:libros#a:empty]]
[[class:libro1#v:{"fecha":1842,"titulo":"Amar","autor":"Juan Diez"}]]
[[class:libro2#v:{"fecha":1922,"titulo":"Odiar","autor":"Diego Diaz"}]]
[[array:libros#a:push#v: [[class:libro1]]]]
[[array:libros#a:push#v: [[class:libro2]]]]

- Declaracion e iteracion sobre objetos: <br> [[array:libros]] <br> <br>
[[iterator:class#array: [[array:libros]]#obj:info#tmpl:
    El autor del libro {<}value{"<}v{-}it_i{>}
    es {<}class{-}it_info.autor{>}
    <br>
]]
- Iteracion con bucle tipo for
[[iterator:simple#range:1..3#tmpl:
    <br>
    Hola Mundo
]]
- Iteracion con bucle tipo for + html
[[iterator:simple#range:1..3#tmpl:
    <br>
    <div style="background: red; font-size:16px; color: white; text-align:center;">
    Hola Mundo
    </div>
]]

```

Clean view

FIGURA 31: EJEMPLO 3 DE PARSEADOR AML